Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 492340, 25 pages
http://dx.doi.org/10.1155/2014/492340

Research Article

Solving a Location, Allocation, and Capacity Planning Problem with Dynamic Demand and Response Time Service Level

Carrie Ka Yuk Lin

Department of Management Sciences, City University of Hong Kong, Hong Kong

Correspondence should be addressed to Carrie Ka Yuk Lin; mslincky@cityu.edu.hk

Received 7 April 2014; Revised 25 June 2014; Accepted 8 July 2014; Published 27 August 2014

Academic Editor: J. J. Judice

Logistic systems with uncertain demand, travel time, and on-site processing time are studied here where sequential trip travel is allowed. The relationship between three levels of decisions: facility location, demand allocation, and resource capacity (number of service units), satisfying the response time requirement, is analysed. The problem is formulated as a stochastic mixed integer program. A simulation-based hybrid heuristic is developed to solve the dynamic problem under different response time service level. An initial solution is obtained from solving static location-allocation models, followed by iterative improvement of the three levels of decisions by ejection, reinsertion procedure with memory of feasible and infeasible service regions. Results indicate that a higher response time service level could be achieved by allocating a given resource under an appropriate decentralized policy. Given a response time requirement, the general trend is that the minimum total capacity initially decreases with more facilities. During this stage, variability in travel time has more impact on capacity than variability in demand arrivals. Thereafter, the total capacity remains stable and then gradually increases. When service level requirement is high, the dynamic dispatch based on first-come-first-serve rule requires smaller capacity than the one by nearest-neighbour rule.

1. Introduction

For many service systems in the public and private sector, the demand sites are often divided into one or more service regions (or zones) to reduce the problem size for service delivery planning. Each service region is typically served by a capacitated facility. A facility could be a physical production/service centre or a collection/redistribution point. In this work, the capacity refers to the number of service units available for dispatch to serve random requests occurring in a region represented in a network of nodes. Examples of such service systems include express delivery, mobile repair service, and emergency systems. This paper examines the combination of a facility location-allocation problem and a queuing problem on a network with response time service level requirement. The planning model includes three types of decisions: (1) number of facilities and locations, (2) allocation of demand sites to each facility, and (3) capacity required at each facility. Response time is measured by the time elapsed between a service request and the arrival of the assigned server. The service level requirement is imposed on the average response time and/or percentage of served requests with response time within a predetermined limit.

The facility location-allocation problem (the first two levels of decisions) involves finding a set of locations of facilities and assigning demand sites to be served by one of these facilities. It has applications in supply chain, logistics, health service planning, and e-commerce for planning of web services provider's facilities and customer allocation [1]. A review of facility location models is given by Arabani and Farahani [2], Farahani and Hekmatfar [3], and a detailed classification in Azarmand and Jamie [4].

Small examples can show that the relationship between number of facilities and the minimum total capacity satisfying the response time requirement is not always a monotonic decreasing function, like that between the number of facilities and maximum travel distance. This is illustrated in two cases of a small static problem with 4 nodes shown in Figures 1 and 2. Assume the travel time between every pair of nodes is equal (15 min.) and each node generates two calls. Suppose

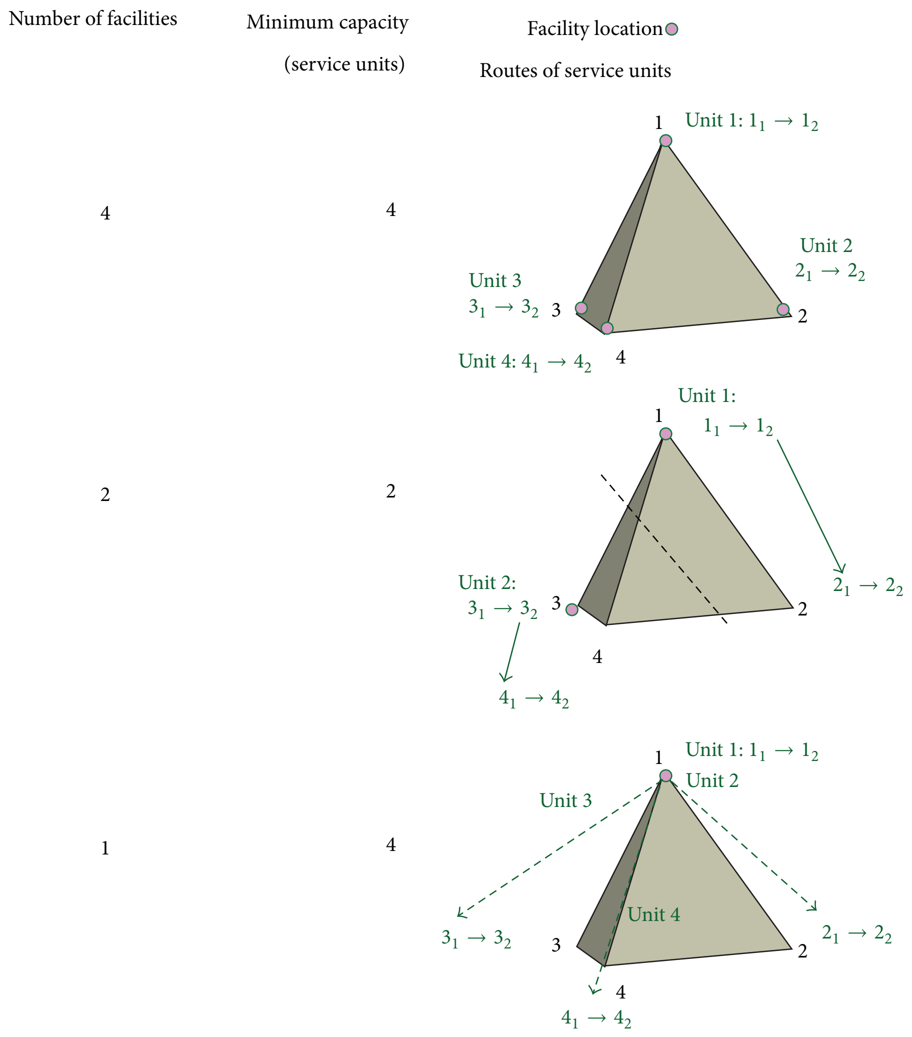


Figure 1: Case 1 (worst case) with all requests arriving at time 0.

the average response time per call should not exceed the travel time between any two nodes. Figure 1 shows the case when all calls arrive simultaneously at time 0 and the route for each service unit with 1 or 2 facilities set up. The scenario of setting up 4 facilities for 4 nodes is trivial. (The three levels of decisions and detailed calculations are shown in Appendix A.) The policy of centralization (1 facility) and complete decentralization (4 facilities) both require larger total capacity than an intermediate policy (2 facilities) that benefits from sharing of capacity (service units) between nodes belonging to the same facility and with appropriate scheduling. Figure 2 shows the same problem but with calls arriving more evenly during an hour at time 0 and the 30th minute in an hour for each node. Again, an intermediate policy (2 facilities) results in the minimum total capacity. If the supplier is capacity-concerned, the intermediate policy is the best. On comparing the total travelling time between the two cases (Tables 6 and 7), Case 1, the more urgent case when all requests are ready at the beginning at each node, results in smaller or the same average travel time. However, the total capacity required is larger than or the same as Case 2 when requests are evenly spread out over time. In both cases, the optimal solution occurs when an additional facility would require a minimum total capacity greater than current capacity in satisfying the response time requirements. For larger problems with more nodes, it is uncertain at what point the total capacity will (first) reach the minimum level. Similar relationship of the three-level decisions with uncertain parameters (arrival time, location, travel time, and on-site processing time) will be explored.

Assumptions on arrival pattern, service time distribution, and travel distance (or time) estimation are crucial to problem formulation and solution. Location-allocation problem with queuing consideration often assumes Poisson distributed demand and exponential service time as in Aboolian et al. [5] and Syam [6]. When total cost is an objective, distance between demand site and facility location is often used to estimate the travelling cost. For rare events like emergency requests, direct travel between facility and demand node is usually assumed to enable quick response. However in operations where servers are mobile units travelling from one node to another to perform service on site,

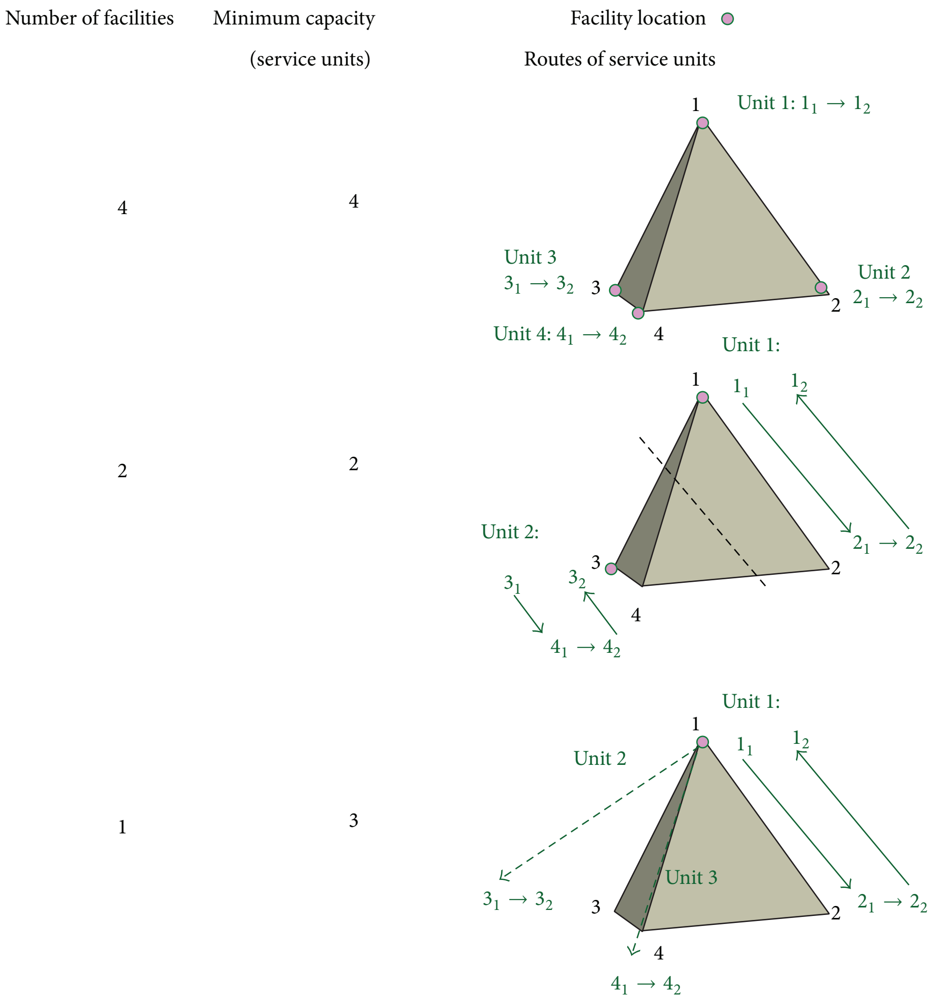


FIGURE 2: Case 2 (average case) with requests arriving at time 0 and the 30th minute in an hour.

the direct distance between demand site and facility would likely overestimate the travelling distance (or time) per trip. Alternative distance measure suggested by Turkensteen and Klose [7] includes a mixed measure basically combining two estimates: (1) the "headway length" estimates the expected distance from demand locations to a central point (e.g., facility) in the area; (2) the "detour length" estimates distance between demand points in the same delivery route. In this work, the response distance limit is used in assigning nodes to facilities (first two levels of decisions) initially. Instead of assuming deterministic distance and fixed decisions, a simulation-based hybrid heuristic (with simulated arrivals, travel time, and on-site work time) is designed to estimate the capacity (third level) and iteratively improve the three levels of decisions.

A "facility" could be static or dynamic. Instead of physical buildings, facilities could be collection points, where items picked up by service units are collected and returned to a high-level sorting/distribution centre. Facilities could also be cross-docking locations, where items from a high-level distribution centre are sorted and distributed to lower level demand sites by service units. Certain logistic systems operate with such a multiechelon structure in which an intermediate layer acts as a facility. Crainic et al. [8, 9] addressed a two-echelon vehicle routing problem where the first echelon represents shipment between a single depot and capacitated satellite stations (or intermediate depots) for consolidation and the second echelon for shipment between satellite stations and customers. In the practice of some courier service, a convenient parking spot could also serve as an intermediate consolidation facility for sorting and distributing mail and is less costly and more dynamic to change location. The cross-docking operation with small facility cost motivates this work and Type 2 experiments (Section 5.2) are designed to model such delivery services.

The main theme of this work explores the case when the entire territory is divided into separate service regions, each served by a facility. The first two levels of decisions correspond to the facility location and demand allocation (service region of facility). The third level decision determines the capacity required for each facility. Within a service region with one or more demand sites, capacity can be shared.

Between service regions, no sharing of capacity is considered in the planning stage to ensure self-sufficiency first in managing demand within a region. Sharing of capacity across service regions will be explored later by allowing a demand site to be assigned to one or more facilities by use of a set-covering model. The assumption of independence, adopted by Brimberg et al. [10], is observed in operating systems with simple management structure or when information sharing between service regions is insufficient. Examples could be found in military application, like field medical units, and one-to-many distribution systems in Turkensteen and Klose [7]. Determining the three levels of decisions helps to answer related questions, such as the following: Given a configuration of facility locations and assigned demand sites, how is it to identify an appropriate response time service level and what is the capacity requirement of a resource? With a given capacity of a resource, how is it to allocate the units to one or more service regions to achieve a higher response time service level?

This paper is organized as follows. Related literature is described in the next section. Section 3 presents the problem statement. Section 4 introduces an initialization stage and a simulation-based hybrid heuristic. Section 5 describes the computational experiments and results. A discussion of the impact of the model assumptions and methodology used on the results is given in Section 6. A summary and final conclusion is given in Section 7 with suggestions for future research.

2. Related Literature

The three-level problem is a combination of well-known single-stage problems: the p-median network location problem, the travelling repairmen problem or minimum latency problem (MLP), and the bounded latency problem (BLP). Even the static version of each such problem is NP-hard. Apart from special cases, like the 1-median problem, the general p-median network location problem which determines the selection of p (uncapacitated) points in a graph and its assigned demand nodes is NP-hard when the objective is to minimize a weighted function of the assignment (Kariv and Hakimi [11]). The k-travelling repairmen problem (k-TRP) determines individual tours for k repairmen to serve all customers with the objective of minimizing the total customer waiting time (or mean arrival time at customers). Even the 1-TRP is NP-hard as shown by Sahni and Gonzalez [12]. Another name for the 1-TRP is the minimum latency problem (MLP) where latency of a customer is a measure of time delay experienced (equivalent to response time here) before his/her service starts. The bounded-latency problem (BLP), a complementary version of MLP, is to find the minimum number of repairmen (equivalent to minimum capacity here) serving all customers such that no customer will wait more than a latency bound (waiting time limit or response time here). The static BLP is first introduced and proved strongly NP-hard in Jothi and Raghavachari [13]. Hence the static version of the current three-level problem is NP-hard, even if any stage has a given known solution. A hybrid heuristic combining optimization, simulation, and search techniques is proposed here.

Related integrated problems include the location and server allocation problem. Extension of median-type location models often involves customers travelling to the closest facility to receive service (e.g., patients to hospitals), where travel distance is assumed deterministic. Aboolian et al. [5] examined a model where the objective is to minimize total system cost (comprising facility fixed cost, server cost, customer travelling, and waiting cost) where the waiting time (and cost) at each facility is approximated by the $M/M/k$ queuing model. An exact algorithm is designed to minimize the server assignment cost for each selected set of facility locations. Heuristics of descent approach and simulated annealing are also proposed to solve larger problems. Other related integrated problems include location-routing problem (LRP) and travelling repairperson location problem. Both consider the location of facilities and routing of servers simultaneously. A common objective of LRP is to minimize the total cost associated with depots, vehicles, and routes. Only a few considered minimizing the total customer waiting time as the objective, such as Averbakh and Berman [14]. The travelling repairperson location problem has a similar objective of minimizing the average response time to an accepted call. Jamil et al. [15] developed a heuristic for a single-server problem in locating an optimal home base, assuming Poisson call arrivals, first-come-first-served queue discipline, deterministic travel times, constant on-scene service times, and a finite capacity queue for waiting calls. The single server can travel directly from one customer site to the next. When no call is in the system, the server would return to the home location and rest for a constant period of time.

Related studies of spatial allocation of resources can be found in field services repair technicians of Hill et al. [16], Chu and Lin [17], and Tang et al. [18]; emergency mobile repair units of Geroliminis et al. [19]; and parcel pickup and delivery service of Wong [20]. Poisson distribution is often assumed to model rare events of failure or accident occurrence rates. Methodologies employed include (state-dependent) queuing model, hypercube queuing-optimization based model, network flow algorithm, and metaheuristics. In Powell [21], service engineers are assigned to jobs in geographical districts represented as a square grid of cells with different given demand. Each engineer is assigned to a cell as the home base, implying that the number of engineers available (service capacity) is simply the given number of cells. A deterministic network flow model (integer variables) was proposed to allocate jobs to engineers, such that in addition to jobs in his own area, an engineer is also allowed to service some requests in his adjacent areas. For a stochastic repair version, Hill et al. [16] developed an approximate state-dependent queuing model to analyze tradeoffs between field service workforce size, territory size, and mean response time. With the territory represented by a rectangular grid, the square-root law proposed by Kolesar et al. [22] and a state-dependent $M/G/s$ queuing model were applied to approximate the expected travel times and expected response times. The analytical expected response time function developed is based on system parameters (including the number of servers) in each of these stochastic problems. Hence, it can be used to determine the minimum number of servers (service capacity) satisfying a given expected response time limit.

Another related application is in medical services. Large-scale emergency response planning involves facility location, assignment of households to facility, and resource allocation of workers of different skills to be stationed at facility as discussed by Lee et al. [23, 24]. For planning of emergency medical systems, the hypercube queuing model by Larson [25] and Larson and Odoni [26] was developed to calculate the mean travel time to incidents, server workload, and number of dispatches per server. The nearest-server dispatching policy is adopted; that is, the server dispatched to a call is always the available unit closest to the call location. Other assumptions of equilibrium state of system, Poisson arrivals, and exponential service times in the $M/M/s$ queuing models were made. In ambulance dispatching problem when several call requests are waiting, Lee [27] designed a centrality policy to dispatch an idle ambulance to a call which is more centrally located with respect to other calls. The average and variation of response time outperformed the well-known nearest-neighbor (NN) policy in the experiments. Beraldi and Bruni [28] examined a location problem of emergency service stations involving assignment of demand points and allocation of vehicles to each station with the objective of minimizing the expected cost. The uncertainty of demand is modelled by a set of scenarios with different probabilities. As an alternative to response time service level, their requirement is based on demand intensity. The decisions made must ensure sufficient vehicles are allocated to a station to fulfil the sum of required vehicles of demand points assigned to it with a given reliability level under each scenario. A demand point can be assigned to one or more stations under each scenario while the location and capacity allocation decisions (number of vehicles) remain invariant under different scenarios. The maximum capacity allowed at each station is an input to be provided by the system planner.

In small package pickup and delivery services, service territory management is one of the key concerns as discussed by Wong [20]. It refers to the territory that a single service provider (service unit) will cover and may have to be adjusted on a daily basis due to workload fluctuations in order to balance workload among different service providers. In the territory planning problem in Zhong et al. [29], the objective is to minimize the total cost of completing the expected workload expressed in terms of vehicle travel time and service time. The main decision for this single-depot problem is the allocation of service areas ("cells") to form a "core area," where the number of core areas corresponds to the minimum number of drivers used, subject to a chance constraint that the assigned workload in the core area for a driver does not exceed his/her maximum working duration with a given probability level. The number of core areas is assumed fixed and estimated by the minimum number of drivers (service capacity) used from historical data. If response times can be incorporated into planning, it can enable a business to increase customer satisfaction and market share under demand uncertainty. In emergency systems, it can reduce casualties. Postal service is considered as a mode of service delivery in emergency response planning as described in Lee et al. [24]. Due to the mail carriers' familiarity with the neighbourhood, they can be used in mass dispensing medicine responding to large-scale biological attack as discussed by Wein [30] and Richter and Khan [31].

Various methods have been applied in solving network design or location problems. These include *integer programming* used in deciding locations, types of service stations, allocation of regions to stations for emergency service system by Coskun and Erol [32], *tabu search* for a multicommodity railway network design problem by Pedersen et al. [33], *metaheuristic* inspired by *variable neighbourhood search* for a multiperiod, multicommodity transportation planning problem by Hoff et al. [34], *genetic algorithms* for a multiobjective location problem by Li and Yeh [35] to maximize population coverage, minimize total transportation costs, and minimize proximity to roads, and *descent* and *simulated annealing* for a location and server allocation problem by Aboolian et al. [5]. This work aims at analyzing three levels of decisions: facility location, demand allocation, and resource capacity requirement. Hence, a hybrid approach combining some of the above concepts will be explored (including integer programming, neighbourhood search with a memory of solutions to avoid recycling).

3. Problem Statement

The problem will be first described followed by a list of assumptions and a mathematical formulation with notations. The problem components consist of a set of demand sites and candidate facility locations represented as m nodes in a network. Facilities could be located at demand sites or as separate nodes. Each facility is equipped with one or more service units travelling to provide service to dynamic requests originated from nodes in its service region. A response time requirement is adopted to ensure the average response time of served requests not to exceed a predetermined limit (R) and a minimum percentage (f_R) of requests served within the limit (R). (If only one condition is required, the other condition can be made redundant by choosing the right parameter value.) The three levels of decisions are (i) determining the number and locations of facilities, (ii) assigning each demand site uniquely to a facility, and (iii) finding the capacity level (number of service units) of each facility. The maximum capacity (number of servers) is assumed unlimited here. In express courier service (Lin et al. [36]), a fixed size of fleet is usually available for servicing requests. However in actual operations, additional backup contract couriers could also be used. (The sum of the two types is considered as the capacity decision here.) In situations when there is a specified maximum capacity limit (like allocation of vehicles to emergency stations in Beraldi and Bruni [28]), the methodology proposed can be easily adapted. The objective is to determine the minimum total capacity in the territory for different number of facilities to fulfill the given response time service level characterized by parameters R and f_R. The curves relating number of facilities and minimum total capacity can provide information for decision-makers in logistic service planning. A list of assumptions made in this research is given in the following.

(1) Each demand site generates dynamic random requests. The mean number of requests at each site and the coefficient of variation are known while the actual number and arrival times are not known with certainty.

(2) Each demand site is to be assigned to exactly one facility.

(3) Each request demands one service unit from the assigned facility.

(4) All requests are treated with the same priority.

(5) The number of facilities to set up is treated as a parameter. (The smallest value is determined in Section 4.1.1. The largest tested value depends on the capacity results in the experiments in Section 5.3.)

(6) The capacity of a facility is expressed by the number of service units to be made available.

(7) All service units begin at a facility with the work duration same as the service session.

(8) The service requirement of a request comprises two stochastic variables: travel time and on-site processing time. The travel times depend on the sequential locations on a server's route which in turn depend on the three levels of decisions (location, allocation, and capacity). The on-site processing time depends on the location and request. The mean time and coefficient of variation of each component are known while the actual values are unknown.

(9) The mean travel time (or distance) between every pair of nodes is assumed symmetrical.

(10) When a service unit completes a service at a customer site, no return to home location (facility) is necessary. Waiting at the current location and direct travel to the next assigned request (i.e., sequential trip travel) is allowed.

The three-level problem with response time requirement and all deterministic, uncertain parameters can be formulated as a stochastic mixed integer program, representing a stochastic bounded latency problem with location decisions. To simplify presentation, it is assumed that a facility could be located at any node i ($= 1, \ldots, m$). (This can be easily adapted to situations when only a subset of nodes can be considered.) A network representation of the problem is presented before the mathematical formulation. Let $G = (V, A)$ be the directed graph with node set V and arc set A. V consists of an artificial source node (s) and a sink node (e) indicating the start and end of each route. Other nodes include location nodes and composite nodes representing a demand site coupled with its request number. Let L_i be the number of requests occurring at demand site i during the service session where both L_i and the actual request arrival time r_{il} of the lth request ($l = 1, \ldots, L_i$) are stochastic parameters that could be generated from the arrival distribution. While a composite node is denoted by il ($l = 1, \ldots, L_i$, $i = 1, \ldots, m$), the simple location node (i) can also be represented as a composite node, $i0$, by assigning the request number $l = 0$. Hence, the arc set A consists of three types of arcs: ($s, j0$) links the source node (s) to a candidate facility node (j) referring to the setup decision; (hk, il) represents two requests served successively on a server's route, with the kth request at node h followed by the lth request at node i; (il, e) indicates the end of a route on completing a request at a demand node, say the lth request at node i. By network construction, the last request served on every route will be linked with the sink node (e).

Notations and Parameters. Consider the following:

T =(deterministic) session duration,

M = (deterministic) an upper limit on the maximum capacity of a facility or a large positive value,

N = (deterministic) number of facilities to set up,

L_i = (stochastic) number of requests from node i ($= 1, \ldots, m$) during session $[0, T]$,

L = (stochastic) sum of requests from all nodes during session $[0, T] = \sum_{i=1}^{m} L_i$,

r_{il} = (stochastic) arrival time of the lth request at node $i, l = 1, 2, \ldots, L_i, i = 1, \ldots, m$,

$t_{i,h}$ = (stochastic) travel time from node i to node h, $i, h = 1, \ldots, m$ (assume $t_{s,i} = t_{i,i}$ =0),

γ_{hk} = (stochastic) on-site processing time of kth request at node h, $k = 1, 2, \ldots, L_h$, $h = 1, \ldots, m$.

Decision Variables. Consider the following:

x_j = (first level decision) 1 if a facility is set up at node j, 0 otherwise, $j = 1, \ldots, m$,

y_{ij} = (second level decision) 1 if demand node i is assigned to facility at node j, 0 otherwise,

$i, j = 1, \ldots, m$,

Z = (third level decision) total capacity or number of service units,

$f_{u,v}$ = flow (number of service units) on arc $(u, v) \in A$,

τ_{il} = server arrival time at the lth request at node i, $l = 1, 2, \ldots, L_i$, $i = 1, \ldots, m$ (assume the start time at a candidate facility i is $\tau_{i0} = 0$, $i = 1, \ldots, m$),

δ_{il} = 1 if the lth request at node i can be served within the response time limit R, 0 otherwise, $l = 1, 2, \ldots, L_i$, $i = 1, \ldots, m$.

Mathematical Formulation (Stochastic Mixed Integer Program). Consider

$$\text{Minimize } Z = \sum_{j=1}^{m} f_{s,j0} \tag{1}$$

subject to

$$\sum_{j=1}^{m} x_j = N \tag{2}$$

$$x_j \le f_{s,j0} \le M \cdot x_j, \quad j = 1, \ldots, m \tag{3}$$

$$\sum_{j=1}^{m} y_{ij} = 1, \quad i = 1, \ldots, m \tag{4}$$

$$y_{ij} \le x_j, \quad i, j = 1, \ldots, m \tag{5}$$

$$\sum_{(u,il) \in A} f_{u,il} = 1, \quad l = 1, \ldots, L_i, \; i = 1, \ldots, m \tag{6}$$

$$\sum_{(u,il) \in A} f_{u,il} = \sum_{(il,v) \in A} f_{il,v}, \quad l = 0, \ldots, L_i, \; i = 1, \ldots, m \tag{7}$$

$$f_{h0,il} \le y_{hh}, \quad l = 1, \ldots, L_i, \; h, i = 1, \ldots, m \tag{8}$$

$$f_{h0,il} \le y_{ih}, \quad l = 1, \ldots, L_i, \; h, i = 1, \ldots, m \tag{9}$$

$$\begin{gathered} f_{hk,il} \le 1 + y_{hj} - y_{ij}, \quad k = 1, \ldots, L_h, \\ l = 1, \ldots, L_i, \quad h, i, j = 1, \ldots, m \end{gathered} \tag{10}$$

$$\begin{gathered} f_{hk,il} \le 1 - y_{hj} + y_{ij}, \quad k = 1, \ldots, L_h, \\ l = 1, \ldots, L_i, \quad h, i, j = 1, \ldots, m \end{gathered} \tag{11}$$

$$\tau_{il} \ge r_{il}, \quad l = 1, \ldots, L_i, \; i = 1, \ldots, m \tag{12}$$

$$\tau_{hk} + \left(T + \gamma_{hk} + t_{h,i}\right) \cdot f_{hk,il} - \tau_{il} \le T, \quad (hk, il) \in A \tag{13}$$

$$\frac{1}{L} \sum_{i=1}^{m} \sum_{l=1}^{L_i} \left(\tau_{il} - r_{il}\right) \le R \tag{14}$$

$$\begin{gathered} \tau_{il} - r_{il} \le \delta_{il} \cdot R + \left(1 - \delta_{il}\right) \cdot T, \\ l = 1, \ldots, L_i, \quad i = 1, \ldots, m \end{gathered} \tag{15}$$

$$\frac{1}{L} \sum_{i=1}^{m} \sum_{l=1}^{L_i} \delta_{il} \ge f_R \tag{16}$$

$$\begin{gathered} x_i, y_{ij} = 0, 1, \quad i = 1, \ldots, m, \\ f_{u,v} = 0, 1, \quad (u, v) \in \{A \mid u \ne s\}, \\ f_{s,j0} \ge 0 \text{ and integer}, \quad (s, j0) \in A, \\ \tau_{il} \ge 0, \qquad \delta_{il} = 0, 1, \quad l = 1, \ldots, L_i, \; i = 1, \ldots, m. \end{gathered} \tag{17}$$

The objective in (1) determines the third level capacity decision to enable the fulfillment of the 2-parameter response time requirement (R and f_R) from all requests. This capacity value is represented by the total outflow (service units) from the source node (s) to facilities in (1). The selection of N facilities is expressed in (2). The facility setup decision is related to the flow variable from the source node to the facility (j) in constraint (3). Constraint (4) requires each facility/demand site to be assigned to exactly one facility. The relationship between the first two levels of decisions, demand assignment and facility setup, is formulated in constraint (5). Every request must be fulfilled exactly once by imposing a unit inflow requirement on the composite demand node (il) in constraint (6). The flow balance constraint for every node (except the source and sink) is formulated in constraint (7). Constraints (8)–(11) relate the flow variables with the demand assignment decisions. Constraints (8) and (9) state that if flow exists from a location node (h) to its first request, then both nodes (h and i) are assigned to the facility at the start location (h). Constraints (10) and (11) formulate similar relationship between two successive requests on a server's route. Both request locations (h and i) must belong to the same facility. They are either assigned together or not assigned at all to a facility (j). Constraint (12) restricts the start service time of a request not to be earlier than its release (or arrival) time. Constraint (13) formulates the precedence relationship between successive requests on a route as well as the subtour elimination constraints. The average response time requirement is enforced by constraint (14). Constraint (15) relates the response time of an individual request with its satisfaction variable (δ_{il}) while constraint (16) requires a minimum fraction f_R of all requests to satisfy the response time limit R. The last constraint (17) declares the type of decision variables and their restrictions. The above formulation is a representation of the three-level decision problem. In the dynamic environment, the stochastic parameter values are only available as the event occurs. The heuristic in the next section is designed to tackle the dynamic problem.

4. Simulation-Based Hybrid Heuristic

The general facility location-allocation problem is NP-hard as shown by Kariv and Hakimi [11]. Under uncertainty in demand calls, locations, and service times, a simulation-based hybrid heuristic is proposed. The initialization stage determines an initial solution for the three levels of decisions, followed by iterative improvement involving optimization, simulation, and ejection and reinsertion search techniques to identify sets of improved feasible solutions.

4.1. Initialization. To obtain the initial location and allocation decisions (first two levels), classical deterministic location-allocation models are solved. For each resulting service region (a selected facility with its assigned demand nodes), the initial capacity value (third level decision) is obtained by solving a deterministic bounded latency problem, simplified from the stochastic formulation (Section 3).

4.1.1. Initial Location and Allocation. The proposed method starts with finding the smallest number of facilities to set up (denoted by N_R) which divides the territory into the same number of separate service regions, each with one facility. This is achieved by imposing the constraint requiring the direct travel time (= distance/average vehicle speed v) between a facility and each of its assigned demand sites to be within the response time limit R. (Note that variation in travel time and demand rates is not considered in the initial solution.) First, define F_{iR} as the set of facility nodes within the direct travel time limit R from a demand site i; that is, $F_{iR} = \{j \mid d_{ji}/v \leq R \text{ and } j \in \{1,\ldots,m\}\}$, where d_{ji} is the direct travel distance from node j to node i $(= 1,\ldots,m)$. The travel distances between pairs of nodes can be found from geographical information systems and the average vehicle speed v from government reports (e.g., Transport Department, Hong Kong [37]). F_{iR} is obtained for demand site i $(= 1,\ldots,m)$ by simply comparing direct distance between node i and every other node. Sets F_{iR}, $i = 1,\ldots,m$, serve as input to finding N_R by the following binary integer program.

Decision Variables:

$x_j = 1$ if a facility is set up at node j, 0 otherwise, $j = 1,\ldots,m$,

$y_{ij} = 1$ if demand site i is allocated to facility at node $j \in F_{iR}$, 0 otherwise, $i = 1,\ldots,m$;

Formulation:

$$\text{Minimize } Z = \sum_{j=1}^{m} x_j \tag{18}$$

subject to

$$\sum_{j\in F_{iR}} y_{ij} = 1, \quad i = 1,\ldots,m \tag{19}$$

$$y_{ij} - x_j \leq 0, \quad j \in F_{iR},\ i = 1,\ldots,m \tag{20}$$

$$x_j, y_{ij} = 0, 1, \quad j \in F_{iR},\ i = 1,\ldots,m. \tag{21}$$

The objective function in (18) determines the smallest number of facilities (N_R) required to reach out to each demand site (i) within the response distance limit ($= R \cdot v$). This limit is fulfilled by the construction of set F_{iR} defined above. Equation (19) assigns each demand site uniquely to a facility. Constraint (20) ensures that a demand site is assigned to an established facility within the distance limit ($R \cdot v$). Constraint (21) declares the decision variables. Thereafter for each input number of facilities, N ($> N_R$), the above step can be skipped. The first two levels of decisions (location and allocation) are initially determined from minimizing the overall workload of the following p-median network location model with $p = N$.

Formulation:

$$\text{Minimize} \quad Z = \sum_{i=1}^{m} \sum_{j\in F_{iR}} \lambda_i t_{ij} y_{ij} \tag{22}$$

$$\text{subject to} \quad \sum_{j=1,\ldots,m} x_j = N \tag{23}$$

and constraints (19), (20), and (21).

The workload in a service region depends on both demand intensities and travel time. The objective of minimizing the overall average (or expected) workload was also adopted in the territory planning problem of Zhong et al. [29]. The total workload in (22) represents the total average travel time taken to serve all requests directly from the facility (j), where λ_i and t_{ij} are the mean request arrival rate from a demand site (i) and mean (direct) travel time between nodes i and j, respectively. (Note that the average on-site processing time,a constant for every unit request, is excluded from (22).) Equation (23) partitions the territory into N ($\geq N_R$) service regions. At this point, the first two levels of decisions, N, $\{x_j\}$ and $\{y_{ij}\}$, have been initially obtained and the service regions are treated as independent problems to provide input to find the third level decision (capacity) and subsequently for further improvement (Section 4.2). To obtain another solution on the curve of number of facilities versus capacity, N ($\geq N_R$) is increased and the algorithm repeats from solving the above model described by constraints (19)–(23).

4.1.2. Lower Bound on Capacity in Service Region. The response time requirement is not considered here when estimating a lower bound on capacity which serves as the initial number of service units (third level decision) for a service region with a facility node (j). This lower bound is obtained by solving a deterministic bounded latency problem (BLP) with side constraints (below). A BLP finds the minimum number of service units required to serve all demand nodes such that each node needs not wait more than a latency bound (equivalent to response time limit). The session duration (T) here defines the latency bound implying all requests to be reached within T. Each demand node is visited at least once by a service unit. Based on assumption (8), the on-site processing time is only request-dependent, but not on the three levels of decisions. The sum of average processing times of all requests, denoted by ρ, will be treated outside the route sequencing decisions as a side constraint (31), representing the supply and demand of working time from all service units. The problem is formulated by a mixed integer network flow model, named as NFM, modified from the stochastic formulation (Section 3). For each given service region (from results in Section 4.1.1), the node set consists of the facility node (j), its assigned demand nodes, and two copies of the facility node representing the source node s and a sink node e. Hence, the travel time from s to each demand node (i) in the region is $t_{si} = t_{ji}$ and $t_{sj}, t_{ii} = 0$. Suppose the total number of facility and demand nodes in the region of j is m_j and let V_j denote the set of such nodes. The arc set, denoted by A, consists of three types of arcs related to nodes h and i in V_j: (s, h) links the source to a facility/demand node h; (h, i) represents two distinct nodes, h and i ($\neq h$), served successively on a server's route; and (i, e) is the arc indicating that i is the last visited node on a route. Then define a subset $A' \subset A$ by excluding all the end of route arcs (i, e) from A, $i \in V_j$.

Decision Variables. Consider the following:

$f_{s,h}, f_{h,i}, f_{i,e}$ = flow (number of service units) on the three types of arc $(s, h), (h, i), (i, e) \in A$,

$a_{hi} = 1$ if arc $(h, i) \in A'$ is used, 0 otherwise,

τ_i = server arrival time at node $i \in V_j$ (assume the start time at the source node = $\tau_s = 0$).

Formulation of NFM. Consider

$$\text{Minimize } Z = \sum_{h \in V_j} f_{s,h} \tag{24}$$

subject to

$$1 \le \sum_{h \in V_j} f_{s,h} \le m_j \tag{25}$$

$$\sum_{(h,i) \in A'} f_{h,i} \ge 1, \quad i \in V_j \tag{26}$$

$$\sum_{(h,i) \in A} f_{h,i} = \sum_{(i,v) \in A} f_{i,v}, \quad i \in V_j \tag{27}$$

$$f_{h,i} \le m_j \cdot a_{hi}, \quad (h,i) \in A' \tag{28}$$

$$\tau_h + (T + t_{hi}) a_{hi} - \tau_i \le T, \quad (h,i) \in A' \tag{29}$$

$$0 \le \tau_i \le T, \quad i \in V_j \tag{30}$$

$$Z \cdot T \ge \sum_{(h,i) \in A'} t_{hi} \cdot f_{h,i} + \rho \tag{31}$$

$$\begin{gathered} f_{s,h}, f_{h,i}, f_{i,e} \ge 0, \quad (s,h), (h,i), (i,e) \in A, \\ a_{hi} = 0, 1, \quad (h,i) \in A', \\ \tau_i \ge 0, \quad i \in V_j. \end{gathered} \tag{32}$$

The objective function in (24) determines the minimum number of service units at the facility (source s) required to complete all requests in a service session. The natural lower and upper bounds of the required service units are imposed on the outflow (service units) from the source (s) to all nodes ($h \in V_j$) in constraint (25). The demand constraint (26) requires at least one service unit to visit demand node $i \in V_j$. Flow balance constraints are formulated in (27) for every node. Subtour elimination constraints are formulated in (28)–(30), where T (session duration) is the latency bound (or maximum waiting time in the session). The sum of average on-site processing times (ρ) is treated in constraint (31), representing the aggregate supply and demand relationship on working time (sum of travelling time and on-site processing time). The supply expression (left side of (31)) helps to determine the objective value—lower bound on capacity. Constraint (32) declares the types of decision variables and their restrictions.

It should be noted that, after obtaining the initial solution for the three-level decisions from this section, the direct travel time limit (= R) between each selected facility and its assigned demand sites needs not be enforced, due to consideration of uncertainty in parameters. (If the facility location is infeasible for its assigned demand sites, this will be reflected in the simulation procedure of the hybrid heuristic in which the response time requirement cannot be satisfied with any increase in service capacity, i.e., no convergence.)

4.2. Iterative Improvement by Simulation and Ejection and Reinsertion Procedure. Apart from the impact of the three levels of decisions, the response time performance also depends on a number of factors, such as geographical characteristics of the service region, time-dependent real-time arrivals, traffic condition, availability of real-time information, and server dispatch logic. To model the uncertainties in call arrival time, travel time, and on-site processing time caused by these factors, a queuing network simulation model SIM (Figure 4) is developed to simulate the operations in each service region. In addition, two choices of simple dispatch algorithms are available in SIM to model the dynamic situation. SIM is used for adjusting the capacity (denoted by s_j) in the service region of the facility node (j) whenever there is under- or overachievement of the response time requirement. To improve the three levels of decisions simultaneously, an ejection, reinsertion procedure is applied iteratively until a termination condition is met. During this iterative procedure, any new feasible/infeasible region evaluated by SIM is recorded to save future simulation time if it reappears in the search. The iterative improvement algorithm is presented in the flowchart in Figure 3 followed by more details of the important components. Notations are first classified and defined as follows.

(i) Solutions: X_0 denotes the initial solution (results of the initialization stage in Section 4.1) and X_1 the first complete solution obtained by applying SIM to find the feasible capacity for each service region in X_0. Let X denote the incumbent solution and X_{new} the new solution generated. After every ejection step when some nodes are ejected from regions, the remaining solution of X is denoted by X_E. Throughout the algorithm, the best found feasible solution is stored in X^* with its total capacity in Z^*.

(ii) Feasible/infeasible service regions: P_F represents the set of feasible service region. In each feasible region, the information recorded is facility node, assigned demand nodes, capacity, server utilization, and average number of served requests per session. P_I represents the set of infeasible service region found during the algorithm. The information recorded for each infeasible region consists only of the facility and demand nodes.

(iii) Ejection pool: E represents the ejection pool holding nodes ejected from an incumbent solution X. The size of E is dynamic and denoted by p. p_{best} records the size p when Z^* was last improved. The dynamic parameter $p_{\max}$ represents the maximum size of E. When there is improvement in Z^* in the present size (p) during a given number of iterations, $p_{\max}$ is increased from p_{best} by a constant.

(iv) Iteration count: dynamic counters include $\text{iter}_{\text{no\_imp}}$, the number of iterations recorded without improvement of Z^*. The maximum value of $\text{iter}_{\text{no\_imp}}$ is denoted by $\text{iter}_{\max}$ (a constant), after which some algorithm parameter values will be changed.

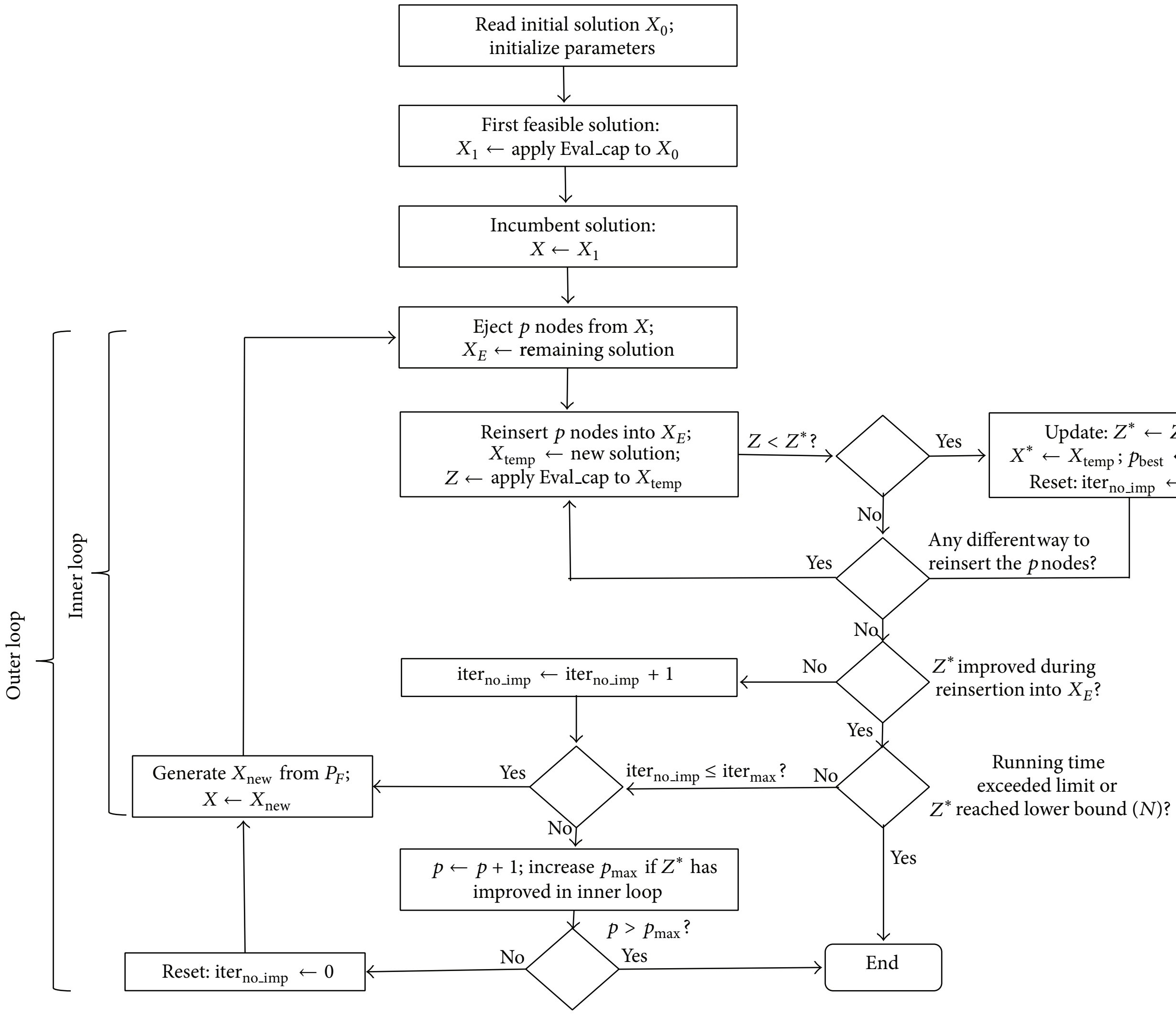

Read initial solution X_0; initialize parameters
First feasible solution: $X_1 \leftarrow$ apply Eval_cap to X_0
Incumbent solution: $X \leftarrow X_1$
Eject p nodes from X; $X_E \leftarrow$ remaining solution
Reinsert p nodes into X_E; $X_{\text{temp}} \leftarrow$ new solution; $Z \leftarrow$ apply Eval_cap to X_{temp}
$Z < Z^*$?
Yes
Update: $Z^* \leftarrow Z$; $X^* \leftarrow X_{\text{temp}}$; $p_{\text{best}} \leftarrow p$. Reset: $\text{iter}_{\text{no\_imp}} \leftarrow 0$
No
Any different way to reinsert the p nodes?
Yes
No
Z^* improved during reinsertion into X_E?
No
$\text{iter}_{\text{no\_imp}} \leftarrow \text{iter}_{\text{no\_imp}} + 1$
Yes
Running time exceeded limit or Z^* reached lower bound (N)?
No
$\text{iter}_{\text{no\_imp}} \leq \text{iter}_{\max}$?
Yes
Generate X_{new} from P_F; $X \leftarrow X_{\text{new}}$
No
$p \leftarrow p + 1$; increase $p_{\max}$ if Z^* has improved in inner loop
$p > p_{\max}$?
No
Reset: $\text{iter}_{\text{no\_imp}} \leftarrow 0$
Yes
Yes
End
Inner loop
Outer loop

Figure 3: Iterative improvement algorithm.

In the iterative improvement algorithm (Figure 3), the initialization step includes reading input data of X_0, initial ejection pool size p, $\text{iter}_{\max}$, running time limit, and setting initial values of variables: $P_F = \phi$, $P_I = \phi$, $X^* = \phi$, $Z^* = \infty$, $p_{\max}$ = initial p + (constant), and $\text{iter}_{\text{no\_imp}} = 0$. The next step finds the first complete feasible solution (X_1) by running an embedded subroutine SIM (Figure 4) in Eval_cap for each region in the initial solution (X_0) (Section 4.1). As the capacity in X_0 is a lower bound, if response time performance (mean and/or percentage achievement) is underachieved, the capacity is increased by one and simulation will repeat until service level is satisfied. Record the current facility location, demand sites, capacity decisions, together with server (or resource) utilization, and average number of served requests from the simulation in the memory array of feasible service regions (P_F). The main concept of the iterative algorithm is to search for more solutions by repeatedly selecting p nodes from an incumbent solution (X) into the ejection pool E and reinserting them back into the remaining solution (X_E) exhaustively using a branch-and-bound procedure. An inner loop of the flowchart applies this concept until there is no improvement on the best total capacity Z^* for a given number of ($\text{iter}_{\max}$) iterations. An outer loop changes the size (p) of E and repeats the inner loop until the maximum size ($p_{\max}$) is reached. The algorithm terminates under any of the three conditions: the running time exceeds a given limit, the lower bound of Z is attained as it reaches the number of facilities (N), or the maximum size ($p_{\max}$) of the ejection pool is reached. Details of important components are summarized as follows.

4.2.1. Arrays of Feasible and Infeasible Service Regions. To save computational effort, whenever the capacity decision of a service region (a facility and assigned demand nodes) has been evaluated by SIM, it will be recorded. Array P_F stores five types of information: the three-level decisions of a feasible region (facility node, assigned demand nodes, and capacity), server utilization (in percent), and average number

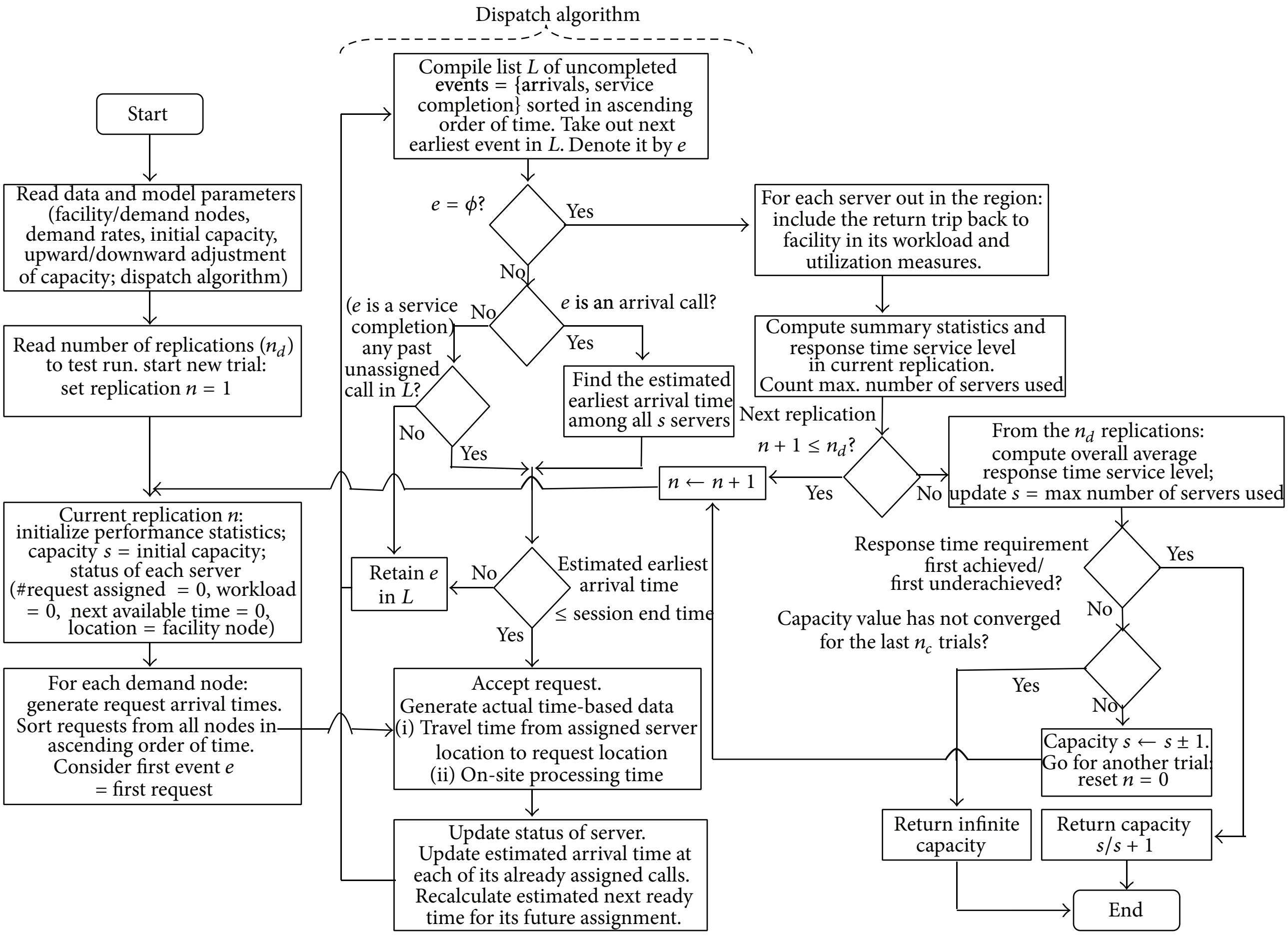

Dispatch algorithm
Start
Read data and model parameters (facility/demand nodes, demand rates, initial capacity, upward/downward adjustment of capacity; dispatch algorithm)
Read number of replications (n_d) to test run. start new trial: set replication $n = 1$
Current replication n: initialize performance statistics; capacity s = initial capacity; status of each server (#request assigned = 0, workload = 0, next available time = 0, location = facility node)
For each demand node: generate request arrival times. Sort requests from all nodes in ascending order of time. Consider first event e = first request
Compile list L of uncompleted events = {arrivals, service completion} sorted in ascending order of time. Take out next earliest event in L. Denote it by e
$e = \phi$?
Yes
No
e is an arrival call?
(e is a service completion) any past unassigned call in L?
No
Yes
No
Yes
Find the estimated earliest arrival time among all s servers
Retain e in L
No
Estimated earliest arrival time ≤ session end time
Yes
Accept request. Generate actual time-based data (i) Travel time from assigned server location to request location (ii) On-site processing time
Update status of server. Update estimated arrival time at each of its already assigned calls. Recalculate estimated next ready time for its future assignment.
For each server out in the region: include the return trip back to facility in its workload and utilization measures.
Compute summary statistics and response time service level in current replication. Count max. number of servers used
Next replication $n + 1 \leq n_d$?
$n \leftarrow n + 1$
Yes
No
From the n_d replications: compute overall average response time service level; update s = max number of servers used
Response time requirement first achieved/ first underachieved?
Yes
No
Capacity value has not converged for the last n_c trials?
Yes
No
Capacity $s \leftarrow s \pm 1$. Go for another trial: reset $n = 0$
Return infinite capacity
Return capacity $s/s + 1$
End

FIGURE 4: Simulation model SIM with first-come-first-serve (FCFS) dispatch algorithm.

of served requests per session. Array P_I only stores the set of facility with its demand nodes not satisfying the response time requirement (i.e., infeasible regions).

4.2.2. Eval_cap. This procedure evaluates the minimum feasible capacity (s) and utilization of a facility in its service region given a starting capacity value ($s = s_0$). If the region already exists in P_F, then the capacity is retrieved for calculations and this subroutine ends. If a new service region is found, the main concept here is to apply the specific dynamic dispatch algorithm on the s service units to complete simulated requests in the region. When the simulation results (in SIM) indicate violation of any response time requirements, s is increased by one and the simulation experiments repeat until the requirements are first satisfied. Capacity s is then returned. The new service region and s will be recorded into P_F together with other relevant information described above. Conversely, if s_0 units are sufficient to fulfil the response time requirement, s (= s_0) will be decreased by one and simulation experiments repeat until the requirement is first violated. Then $s+1$ will be returned as the capacity of this new region to be stored in P_F. Note that Eval_cap may involve recomputing the new facility node if the original facility node has been ejected as one of the p nodes in the iterative improvement algorithm. The testing of new facility node is in increasing order of workload (= sum of {average travel time from the tested facility to node × demand rate of node} over all nodes in the region). If no capacity (s) can satisfy the response time requirement, the nodes of this region will be stored in P_I and infinite capacity is returned.

4.2.3. Generate New Incumbent Solution. In the iterative improvement procedure, a new solution X_{new} is generated to diversify the search. It is obtained by solving a set-partitioning model on P_F with an additional lower bound constraint $Z \geq \lfloor r \cdot Z^* \rfloor$. This imposes a gap between the objective value Z (total capacity) of the new solution and Z^* by a random factor r, where $1 \leq r \leq 2$. If the lower bound, $\lfloor r \cdot Z^* \rfloor$, is too large that there is no solution, it will be relaxed to half the value of Z^* and the current lower bound (i.e., $(Z^* + \lfloor r \cdot Z^* \rfloor)/2$). The set-partitioning model with the revised lower bound constraint is rerun until a feasible solution, denoted by X_{new}, is obtained.

4.2.4. Select (p) Nodes from the Incumbent Solution into Ejection Pool E. The rationale behind this is to allow changes to an incumbent solution X by ejecting p nodes through two procedures.

(i) First, select a facility (j) randomly from X based on the total utilization (= number of service units × server utilization percentage) of facilities in a way such that a facility with higher total utilization has larger probability of being selected.

(ii) Randomly select a demand node in facility j based on its proximity to other facility nodes in a way such that a node being closer to other facility nodes is more likely to be selected.

(iii) Repeat the above until p nodes are selected into E.

4.2.5. Perform Reinsertion. The p nodes in E are reinserted in different ways back into service regions of the remaining solution (X_E) by the branch-and-bound procedure.

(i) For every node (i) in E, find its distance to each facility node in X_E. Sort the distances in increasing order to create an ordered list, denoted by L_i, of size N. Repeat for every other node in E. The joint list formed is $L_1 \times L_2 \times \cdots \times L_p$.

(ii) In each set of insertion, node i in E is inserted into a facility in list L_i ($i = 1, \ldots, p$) until each of the p nodes is assigned a facility. This is followed by capacity evaluation of each affected facility (with its assigned nodes) by applying procedure Eval_cap.

(iii) When a partial or complete solution ($\leq N$ service regions) is evaluated with sum of capacity $> Z^*$, this set of inferior reinsertion is discarded. Accordingly when a feasible complete solution obtained has smaller total capacity $Z < Z^*$, Z^* and X^* are updated.

4.2.6. Size of Ejection Pool. Whenever the best objective value (Z^*) is improved, the current size (p) of E will be recorded as p_{best} and the ejection procedure is allowed to eject more nodes from X by increasing the maximum size ($p_{\max}$) of E from p_{best} by a constant parameter.

4.2.7. SIM (Simulation Model). SIM is a discrete-event queuing network simulation model with spatial requests arriving dynamically in a given service region. The objective is to find the minimum capacity in the given service region satisfying the response time requirement. It is embedded within Eval_cap. The advantage of using simulation approach is the flexibility in testing different variable factors (and their interactions) to generate a distribution of outputs under a given response time requirement (R and f_R). Unlike static problems where all information is available at the beginning, the data in the dynamic problem (e.g., request arrival time, next available time of server) will only be known when the event occurs. For a given service region, the variables considered include the number of requests, their arrival times, location, travel time and on-site processing time. In addition, two simple dispatch algorithms are applied to examine their impact on the heuristic performance. Figure 4 shows the logical design of SIM with the dispatch algorithm first-come-first-serve (FCFS). An alternative based on the nearest-neighbour rule (NN) is available for dispatching a server to a nearby request after its actual service completion. The pseudocode of NN is shown in Algorithm 1.

The *variation* in each time-based variable (arrival time, travel time, and on-site processing time) is modelled by a normal distribution characterized by a given mean and standard deviation. For travel time data, given that the mean travel time (= distance/average vehicle speed v) from node i to node j is t_{ij} and coefficient of variation is c_T, the travel time will be simulated from a normal distribution with mean t_{ij} and standard deviation $c_T \cdot t_{ij}$. To avoid extreme values being simulated, a lower bound of one-half and an upper bound of three times the mean t_{ij} are imposed on the simulated travel time. As requests are not necessarily failures or rare events (often approximated by Poisson arrivals), the interarrival time of request at each demand node (i) is approximated by a normal distribution with mean arrival rate λ_i (or mean interarrival time of $1/\lambda_i$)and coefficient of variation c_a of interarrival time (i.e., standard deviation is $c_a \cdot 1/\lambda_i$). Requests generated from all demand nodes are then sorted in ascending order of arrival times for dispatch. The service time of a request comprises the sequence-dependent travel time and on-site processing time. Similarly, the on-site processing time will be simulated from a normal distribution with given mean γ and standard deviation $c_\gamma \cdot \gamma$, where c_γ is the coefficient of variation. SIM will be run for a predetermined number of (n_d) replications over which the performance statistics of average travel time, average response time, percentage achievement of response time limit (R), server utilization (percentage), and average number of served requests are collected. *Adjustment of capacity* for a service region in the simulation model is necessary when the response time requirement is under- or overachieved. Typically the initial capacity in each region (from objective in (24)) is likely a lower bound. The adjustment is set upwards in incremental step of 1 service unit. When evaluating the capacity of the remaining solution (X_E) after ejection of nodes, the adjustment is downwards in steps of −1 service unit even if the input capacity is feasible. The adjusted capacity will act as input to SIM for repeating another set of n_d replications until the response time requirement is first achieved. For upward adjustment, the first feasible capacity is then returned. As for downward adjustment of capacity, the stopping condition is the first occurrence of underachievement of the response time requirement. Hence, one additional unit to the current capacity will be returned as the minimum capacity required. Note that a given service region could be infeasible. This is realized when the capacity does not converge for the last n_c trials (where each trial consists of n_d replications). In Type I experiments assuming unit request per node (Section 5.1), the value of n_c is set to be the number of nodes in the region as it takes at most n_c trials to adjust the capacity to its upper (or lower) limit. Otherwise, n_c can be an input parameter. When no convergence occurs, the infeasible facility and demand nodes will be stored in P_I and infinite capacity is returned from SIM.

4.2.8. Dispatch Algorithms (Dynamic). The motivation behind the choice of the two dispatch algorithms is to mimic the dispatch logic of some practical systems as much as possible

```
For each demand node in the given region: Generate request arrival times
Sort requests from all nodes in ascending order of time into a list $L$
Initialize: Let current time $\tau = 0$; $e$ = first event in list $L$ with event time $\geq\tau$
Repeat
    If $e$ is an arrival request Then
        If all servers are busy Then $s_e \leftarrow \phi$ (no server assigned);
        Else find the estimated earliest arrival time, $t$, among all available servers.
            Let $s_e$ be the server which can arrive the earliest (estimated) at $e$
        Endif
    Else {$e$ is an actual service completion}
        Let $s_e$ = server available after completing $e$
        If there is any unassigned request(s) on or before current time $\tau$ Then
            Find the unassigned request nearest to the current location of $s_e$
            Estimate the arrival time, $t$, of $s_e$ to the request location
        Else reset $s_e \leftarrow \phi$
        Endif
    Endif
    If $s_e \neq \phi$ and $t \leq$ session end time Then
        Assign request to server $s_e$
        Generate the actual travel time and on-site processing time to determine the actual completion time $t_e$
        Store $t_e$ and related information (request, location, server $s_e$) in list $L$
    Else retain $e$ and related information (location, server $s_e$) in list $L$ for later assignment
    Endif
    Sort $L$ in ascending order of time
    Advance $\tau \rightarrow$ the first event time ($\geq\tau$) in list $L$. Let $e$ denote the event
Until a termination condition* holds.
*3 termination conditions: (i) $\tau$ exceeds the session end time (ii) all requests are assigned or
(iii) all requests have arrived and $\tau$ reaches the completion time of the last event in list $L$
```

Algorithm 1: Pseudocode of nearest-neighbour (NN) dispatch algorithm.

(without involving too much computational time as simulation is adopted). Examples include the online taxi automation system mentioned by Mandle et al. [38] which primarily focused on reaching individual requests in the shortest possible time to enhance customer satisfaction. Requests are prioritized in a first-come-first-serve manner to be assigned to the nearest taxi. In emergency ambulance dispatch, a common rule is to send the closest unit to the request site. Various researchers studied other dispatch strategies. Bandara et al. [39] incorporated call urgency into their proposed dispatch heuristic which assigns the nearest available ambulance for Priority 1 calls and the less busy ambulance for Priority 2 calls. Results from simulation experiments reported an increase in patient survivability, decreased average response time, and higher percentage of Priority 1 calls served within 10 minutes. As requests have equal priority here (assumption (4)), they would all be treated like Priority 1 calls. The main difference between the two dispatch algorithms is customer-based (in FCFS) or server-based (in NN) and the use of estimated or actual service completion time information in dispatching servers to requests. In FCFS (Figure 4), requests prioritized by their arrival order are assigned to the server with the estimated earliest arrival time. This is calculated by the sum of the estimated earliest completion time (of already assigned requests) of the server and the mean travel time from his last request location to the request being considered. The completion time estimates are updated whenever an actual completion occurs to facilitate future assignments. On the other hand, the request arrival order may not be respected in NN (Algorithm 1). Unassigned requests will be stored in a list (L) and whenever an actual service is completed, the server will be dispatched to the nearest unassigned request. Both dispatch methods will use the mean travel time (t_{ij}) between the two locations in estimating the arrival time of a unit at the request location. When the estimated earliest arrival time exceeds the session end time, the request will not be accepted temporarily and will remain in the list for future assignment to possibly another nearby server. Otherwise, the request will remain unserved in the session.

4.3. Impact of Sharing Capacity. Sharing capacity will allow a demand node to be served by one or more facilities. To explore possible advantage, a classical set-covering model (Appendix B) with an additional constraint to select N facilities (assumption (5) in Section 3) is applied to the final pool of feasible regions (P_F) at the end of the hybrid heuristic. The basic (binary) decisions $\{x_j, j \in P_F\}$ involve selecting N service regions in P_F with minimum total capacity (or other objective criteria) such that each demand node in the territory is *covered* by at least one facility. (If an alternative set of N service regions (or facilities) is to be generated, one can simply add one more constraint $\sum_{j\in\Omega} x_j \leq N-1$ to forbid the recent set Ω to be reselected.)

5. Computational Experiments and Results

Two types of experiments are performed in testing the simulation-based hybrid heuristic. The first type simulates disaster outbreak with multiple requests occurring in a short time interval. Twelve data sets of size from 29 to 262 cities are selected from the Travelling Salesman Problem library [40], assuming unit request per node. Certain variables (interarrival time distribution, travel time variability, and dispatch algorithm) are tested with two alternatives each to examine their impact on the total capacity. The second type of experiments simulates the express delivery service environment where arrivals span over a longer time period and nodes in the service network have different demand intensities, like clustered customer nodes in urban cities.

5.1. Type 1 Experiments: Requests Arriving Early in a Short Interval. The data sets from the Travelling Salesman Problem (TSP) [40] provide the distances in the network. Each node here is assumed to have one unit request. In an extreme case, all requests are released at the beginning of the service session. With no variability in travel time and on-site processing time, the problem would be deterministic. In reality, the request calls arrive dynamically over a short interval; travel times and on-site processing times are uncertain. Hence, results from a deterministic problem are used as a reference for comparison with the hybrid heuristic solving the dynamic problem. The deterministic problem of finding the minimum number of servers such that each request needs not wait more than a time limit (latency bound) has been introduced as the bounded latency problem (BLP) for a given facility and its demand nodes (Section 2: Jothi and Raghavachari [13]). When facility decisions are unknown, the problem is named here as the BLP with location decisions (Appendix C)—a simplified deterministic version of the stochastic mixed integer program (Section 3) with unit request per node and all requests available at time 0. The computational time required for solving this NP-hard problem is significant for problems of medium to large size. Hence, it will be allowed more running time than the hybrid heuristic. After observing the performance of both methods in preliminary experiments, the maximum time limit allowed for the mixed integer program (Appendix C) would be 1,800 CPU seconds and half of that for the hybrid heuristic, that is, 900 CPU seconds. Beyond these limits, the objective function shows little improvement. Model NFM, providing the initial capacity for each region (Section 4.1.2), is given only 30 CPU seconds, while allowing more time ($900 - 30 = 870$ CPU seconds) for the iterative improvement procedure in the hybrid heuristic.

Type 1 experiments simulated the dynamic problem in which the hybrid heuristic is applied with two dispatch algorithms separately. Due to incomplete information, results from a dynamic problem would not be better than an optimal solution of an equivalent deterministic problem. Nevertheless, it is difficult to solve an NP-hard problem exactly especially for large problems. In general, it is observed that the (optimal) result of total capacity from the mixed integer program would provide an upper bound to the heuristic value, unless there is large variability in some parameters. (When there are few facilities and variability in travel time is moderately high ($c_T = 1$), simulation results show that, even with a large capacity, a high response time requirement (e.g., $f_R = 95\%$) could not be fulfilled (Figure 7, when $N = 4$). This insight will also be illustrated in the small static example in Figures 1 and 2 by comparing the impact on total capacity in varying travel time versus demand intensity.) For simplicity, the on-site processing time is assumed to be zero here. To model the dynamic situation, the following three factors and two alternatives of each are tested in the hybrid heuristic, giving a total of 8 versions for each TSP data set for each value of N (Tables 1, 2, 3, and 4):

(i) interarrival time distribution: uniform distribution; normal distribution with $c_a = 0.5$,

(ii) variability in travel time: $c_T = 0$; normal distribution with $c_T = 0.5$ ($c_T = 0$ implies constant average travel time between nodes),

(iii) dynamic algorithms explained earlier (Section 4.2): FCFS and NN.

5.1.1. Heuristic Parameters. The choice of parameter values and its variability affect the rate of change of capacity with the number of facilities. When the response time requirement is tight (small R and/or large f_R), larger capacity saving is observed when facilities are added to its minimum feasible value (Figures 5–7), especially when travel time has large variability. To select the value of R, the distances are converted to time by assuming a vehicle speed of $v = 100$ km/hour. As each request should not wait more than R, the value of R is chosen as a calculated factor, $\alpha \cdot \min_i\{\max_j\{t_{ij}\}\}$, from the data set, where α is a constant taken from $[0.6, 1]$ and t_{ij} is the average travel time from node i to node j. The factor α reflects the closeness of demand nodes from one or more facilities within the time limit R. To simulate calls arriving early in a short interval in the dynamic environment, the call arrival times are generated from the normal distribution with mean at $R/2$ and the largest arrival time bounded by R. The session duration is set to be $T = 2R$ to allow late-arriving calls to be able to satisfy the response time requirement. Results are compared with the alternative uniform arrival distribution with simulated requests arriving over the same interval $[0, R]$ and with the same T. To ensure that *almost* all calls would be served for comparison with results from the mixed integer program (Appendix C), a high percentage fulfillment (f_R) of 95% is chosen, as well as for the percentage of calls required to be served within the session. In both Type I and II experiments, the number of replications in SIM is chosen to be $n_d = 25$ and initial ejection pool size $p = 2$. When $p_{\max}$ is updated (from p_{best} + constant), the incremental constant = 3 and $\text{iter}_{\max} = 7$.

5.2. Type 2 Experiments: Requests Span over Time with Higher Variability in Data. The operating environment of express delivery services allowing multiple requests per demand node is simulated here. Customer-centred response time performance will help increase revenue. As the problem size is large, only the hybrid heuristic with different input

Table 1: Total capacity of mixed integer program (MIP) and simulation-based heuristic on TSP data sets: bays29, att48, and eil51.

Total capacity	bays29 (29 nodes, $R = 153$)					att48 (48 nodes, $R = 840$)					eil51 (51 nodes, $R = 30$)				
Number of facilities (N)	1	2	3	5	8	1	3	5	7	9	1	3	5	7	9
MIP	12*	9	8	8	8*	12	9	10	10	9*	13	12	12	12	11
Heuristic: arrival(c_a), c_T, dispatch															
Uniform(-), 0, FCFS	9	8	8	7	8	10	8	8	9	9	11	10	10	9	9
Normal(0.5), 0, FCFS	10	9	9	9	8	12	9	10	10	9	12	11	11	11	11
Uniform(-), 0, NN	12	9	8	7	8	15	8	8	8	9	13	10	9	9	9
Normal(0.5), 0, NN	14	11	10	9	8	17	10	8	9	9	14	12	11	10	10
Uniform(-), 0.5, FCFS	#	10	10	8	9	#	10	10	10	10	15	11	11	10	11
Normal(0.5), 0.5, FCFS	#	11	10	10	10	#	11	11	11	11	17	13	13	13	12
Uniform(-), 0.5, NN	#	13	11	9	8	#	10	9	9	9	26	12	11	10	10
Normal(0.5), 0.5, NN	#	16	13	10	9	#	11	10	10	10	31	14	12	11	11

*Optimal; #no convergence in capacity.

Table 2: Total capacity of mixed integer program (MIP) and simulation-based heuristic on TSP data sets: berlin52, pr76, and gr120.

Total capacity	berlin52 (52 nodes, $R = 570$)					pr76 (76 nodes, $R = 7116$)							gr120 (120 nodes, $R = 240.66$)						
Number of facilities (N)	1	3	5	9	11	1	3	5	8	10	11	12	2	6	10	14	18	22	23
MIP	11	11	10	11	11*	16	17	16	17	16	13	16	—	—	39	29	29	27	35
Heuristic: arrival(c_a), c_T, dispatch																			
Uniform(-), 0, FCFS	10	9	9	9	11	12	12	11	11	11	11	12	22	21	21	20	18	22	23
Normal(0.5), 0, FCFS	11	10	10	10	11	13	13	13	13	12	12	13	25	22	22	23	23	23	24
Uniform(-), 0, NN	12	8	8	9	11	17	11	11	10	11	11	12	24	19	19	20	19	22	23
Normal(0.5), 0, NN	15	10	9	10	11	19	14	11	11	11	11	12	30	22	22	21	21	22	23
Uniform(-), 0.5, FCFS	15	10	11	10	11	14	14	13	12	12	12	13	29	23	24	24	22	22	24
Normal(0.5), 0.5, FCFS	16	11	11	10	12	27	15	14	14	14	13	15	33	26	25	28	26	26	28
Uniform(-), 0.5, NN	28	10	9	10	11	36	14	13	12	11	11	12	62	22	22	20	22	22	24
Normal(0.5), 0.5, NN	33	11	10	10	11	48	15	14	12	12	11	12	57	24	25	24	23	23	23

*Optimal; —: no solution.

Table 3: Total capacity of mixed integer program (MIP) and simulation-based heuristic on TSP data sets: ch130, kroA150, and sil175.

Total capacity	ch130 (130 nodes, $R = 288$)						kroA150 (150 nodes, $R = 1326$)						sil175 (175 nodes, $R = 182$)					
Number of facilities (N)	2	7	10	13	16	19	1	4	8	12	17	20	7	10	15	20	25	30
MIP	—	106	35	40	42	42	54	—	44	51	56	50	—	—	—	—	—	115
Heuristic: arrival(c_a), c_T, dispatch																		
Uniform(-), 0, FCFS	19	17	17	17	18	19	23	18	18	17	17	20	59	56	55	54	54	53
Normal(0.5), 0, FCFS	20	20	20	20	19	21	25	20	19	21	19	21	68	67	66	65	65	67
Uniform(-), 0, NN	20	16	15	15	16	19	26	17	16	16	17	20	66	62	58	58	58	56
Normal(0.5), 0, NN	22	17	17	16	16	19	27	18	16	16	17	20	77	74	71	72	70	68
Uniform(-), 0.5, FCFS	23	20	19	20	19	20	43	21	20	21	18	20	109	88	82	76	76	74
Normal(0.5), 0.5, FCFS	24	22	23	24	20	22	51	23	21	23	20	22	123	103	92	88	85	86
Uniform(-), 0.5, NN	26	18	17	16	16	19	#	18	17	17	18	20	112	90	78	72	70	67
Normal(0.5), 0.5, NN	27	19	18	18	17	20	#	19	17	19	17	21	125	105	91	85	82	83

—: no solution; #no convergence in capacity.

factors will be run for comparative analysis with no absolute comparison from the mixed integer program (Appendix C). The data and parameters came from three sources of delivery service.

(i) The first source provides locations and demand data in the network from real-life data of a local delivery service with customers distributed in 89 housing estates (demand sites) and 5 candidate depots on Hong

Table 4: Total capacity of mixed integer program (MIP) and simulation-based heuristic on TSP data sets: d198, tsp225, and gil262.

Total capacity	d198 (198 nodes, R = 983)				tsp225 (225 nodes, R = 113)						gil262 (262 nodes, R = 49)						
Number of facilities (N)	4	5	6	10	2	5	10	15	20	30	4	10	20	25	30	35	40
MIP	—	56	—	52	—	—	—	84	98	81	—	—	—	—	95	108	124
Heuristic: arrival(c_a), c_T, dispatch																	
Uniform(-), 0, FCFS	12	12	12	14	34	29	27	31	29	30	42	40	37	39	38	40	41
Normal(0.5), 0, FCFS	13	14	15	16	37	32	31	31	34	33	47	45	43	43	46	46	47
Uniform(-), 0, NN	9	9	10	11	40	27	25	25	24	30	47	39	36	35	39	37	40
Normal(0.5), 0, NN	10	9	11	12	46	29	27	29	27	30	55	43	38	40	39	39	41
Uniform(-), 0.5, FCFS	14	14	13	15	50	34	31	31	32	32	54	46	45	43	43	43	45
Normal(0.5), 0.5, FCFS	15	14	14	16	56	37	35	35	37	35	60	50	50	50	51	49	51
Uniform(-), 0.5, NN	9	10	10	12	#	31	26	30	25	30	89	43	37	39	38	37	41
Normal(0.5), 0.5, NN	10	10	10	12	#	35	27	31	29	32	98	45	42	42	40	43	44

—: no solution; #no convergence in capacity.

Kong island resulting in a total of 94 nodes [41]. The geographical characteristics consist of densely populated clustered districts. Pairwise travel distances were estimated by a geographical information system. An average vehicle speed of 20.1 km/hour provided by Transport Department, Hong Kong [37], helps to convert distances into travel times. The average daily demand volume is 21,090 from all demand sites (including both residential and business customers). Data set from this source are made available on internet [42], with mean daily volume of node i (= $1,\ldots,94$) denoted by λ_i'.

(ii) The second source, an international express delivery company described in Lin [43], helps to approximate the express demand. It has 550 customer locations in Hong Kong while the geographical location is not disclosed. To approximate the express demand in mail delivery (first source), an adjustment factor of 550/21090 is adopted. Assuming that a total of 550 express requests occur on an average day of 10 work hours, demand node i has an estimated mean request arrival rate (per minute) of $\lambda_i = \lambda_i' \times 550/21090/(10$ work hours per day $\times$ 60 minutes). Accordingly, the interarrival time distribution has mean $1/\lambda_i$ and standard deviation $c_a \cdot 1/\lambda_i$. A minimum time gap of 10 seconds is imposed between the arrival of successive requests and a minimum of 5 minutes on the travel time to the next request.

(iii) The third source, a Taiwan express delivery company in Lin et al. [36] together with the second source in Hong Kong [43], provides certain operational information in the express delivery service, including the average on-site processing time (γ) of 5 minutes. A minimum of 2 minutes and maximum of 30 minutes are assumed here. A workday actually consists of two half-day sessions: 3.5 hours and 6.5 hours, respectively, with a lunch break of 1.5 hours. The experiments here will focus on the longer afternoon session (6.5 hours), assuming independent work sessions. Unaccepted calls arriving near the end of the session will be assumed lost.

The scenario with a moderate variability in time-based parameters is tested by setting $c_a = c_T = c_\gamma = 1$ (with upper and lower bounds imposed on simulated values mentioned above). Response time service level is tested from low to high requirement. The mean response time limit R assumes some convenient values of 90, 60, and 30 minutes. Two sets of percentage requirements are adopted for each R: f_R = 50% and 90%, respectively. To compare fairly between the two dynamic dispatch algorithms (FCFS and NN) and ensure that both serve similar number of requests, the average percentage of served requests recorded in FCFS is enforced as a lower bound in NN when requirement is low (f_R = 50%). When requirement is high (f_R = 90%), the average percentage of served requests must be at least 95% in both dispatch algorithms. The runtime of different components in the hybrid heuristic is chosen after initial testing. As in Type 1 experiments, the hybrid heuristic has a running time limit of 900 CPU seconds in which 30 CPU seconds are allowed in Model NFM for finding the initial capacity for each region (Section 4.1.2).

In both Type 1 and 2 experiments, the number of facilities (N) tested starts from the smallest feasible value, assuming that the mean direct travel time between a facility and each demand node is within R (Section 4.1.1). The subsequent N values are chosen when the total capacity results can reflect significant change in the curve. The largest N is chosen when the heuristic capacity reaches the lower bound (N) or shows an increasing trend (i.e., worse). All algorithms are coded in Visual Basic.NET 2005 version and the experiments are performed on a Pentium 4, 2.5 GHz processor. IBM ILOG CPLEX 12.5 is used to solve the small static 4-node example in Figures 1 and 2 (mathematical formulation in Section 3), optimization subproblems in the hybrid heuristic, and the deterministic problem (Appendix C) for comparison of results in Type 1 experiments.

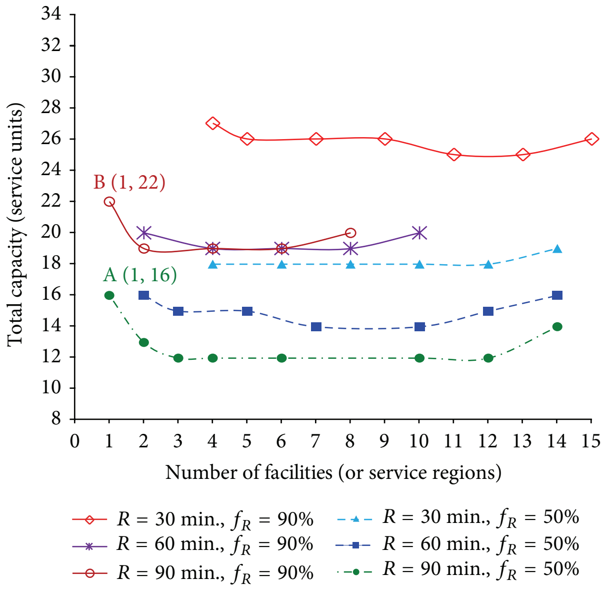


FIGURE 5: Relationship between total capacity and number of facilities using FCFS dispatch algorithm under moderate variability ($c_a = c_T = c_\gamma = 1$).

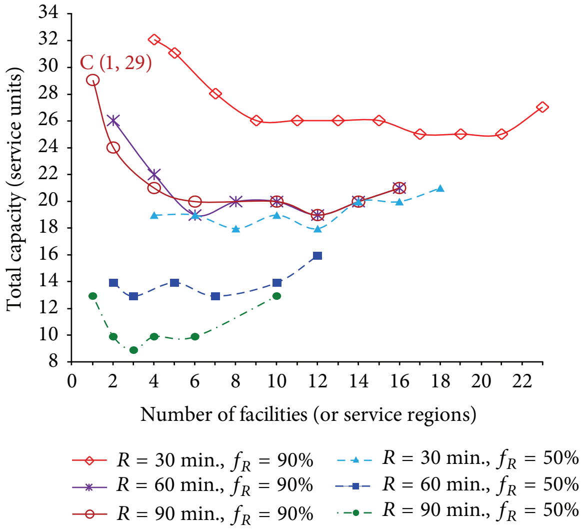


FIGURE 6: Relationship between total capacity and number of facilities using NN dispatch algorithm under moderate variability ($c_a = c_T = c_\gamma = 1$).

5.3. Test Results. Tables 1–4 show the results for Type 1 experiments based on the 12 TSP data sets ranging from 26 to 262 nodes. The mixed integer program (Appendix C) can only reach optimality in 4 instances with fewer than 100 customer nodes (Table 1: data set bays29 with $N = 1, 8$; att48 with $N = 9$; and Table 2: berlin52 with $N = 11$).

In the first five data sets (Tables 1 and 2) with fewer than 100 nodes, the mixed integer program and hybrid heuristic produce comparable values in total capacity. When variability in parameters is small (uniform arrival distribution or $c_T = 0$), the heuristic produces lower capacity value than the mixed integer program, apart from instances when number of facilities (N) is at its smallest value (i.e., $N = 1$ or 2 in Tables 1 and 2). When N is small and travel time variability measure $c_T = 0.5$, no convergence in capacity or very large value was experienced in several data sets (e.g., Table 1: bays29 and att48), implying impossible achievement of the response time requirement. The decrease in total capacity is more significant when N increases from the smallest feasible value to the next (e.g., Table 1: data set bays29 when $N = 1 \rightarrow 2$; att48 when $N = 1 \rightarrow 3$). When problem size increases beyond 100 nodes (Tables 2–4), it becomes increasingly difficult for the mixed integer program to obtain a feasible or improved solution. The relationship of the capacity versus N is irregular in the MIP

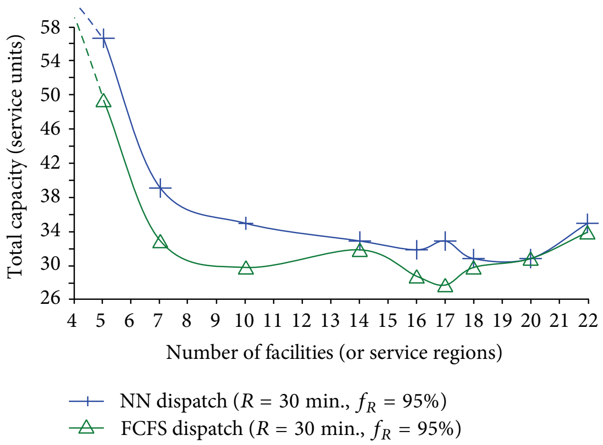


Figure 7: High service level curves ($R = 30$ min., $f_R = 95\%$) for the FCFS and NN dispatch algorithms under moderate variability ($c_a = c_T = c_\gamma = 1$).

results. For larger problems of over 180 nodes (Table 4), only feasible solutions can be obtained from the mixed integer program when N is large.

Some insights are drawn from the experimental results (Tables 1–4) of the hybrid heuristic.

(i) Initially, there is a decreasing trend of total capacity with decentralization of facilities as observed for each scenario tested on a data set. Thereafter, the total capacity levels with only small variation due to randomness. Beyond a certain limit, capacity would gradually increase as each additional facility will require a minimum amount of resource (e.g., 1 service unit).

(ii) The higher the variability of parameters (interarrival time, travel time), the larger the total capacity required. (Compare capacity results of scenario Uniform arrival rate versus Normal ($c_a = 0.5$) while the other two factors remain the same. Similarly, compare scenario $c_T = 0$ versus $c_T = 0.5$.)

(iii) The variability in travel time increases capacity more than variability in interarrival time, particularly when only a few facilities are set up. (For small N values, compare change in capacity between two pairs of scenario: (1) scenario Uniform arrival rate to Normal ($c_a = 0.5$), both with $c_T = 0$; and (2) $c_T = 0$ to $c_T = 0.5$, both with Uniform arrival rate. For instance, in Table 4, data set gil262, at the smallest $N = 4$:

Based on FCFS dispatch, (1) $47 - 42 = 5$; and (2) $54-42 = 12$. Hence, capacity change in (1) < (2).

Based on NN dispatch, (1) $55 - 47 = 8$; and (2) $89 - 47 = 42$. Hence, capacity change in (1) < (2).

In both types of dispatch, this insight is valid. The same is observed for other data sets when N is small.)

Another illustration is on the small static 4-node example in Figures 1 and 2. We explore the change in capacity if the actual demand is doubled, versus the actual travel time takes twice as long, while other parameters remain unchanged. The scenario of duplicating the demand (number of requests per node increases from 2 to 4) versus doubling the travel time between every two nodes (from 15 to 30 minutes) is compared. The optimal (minimum) capacity satisfying the same response time requirement (average response time ≤ $R = 15$ minutes) is shown in Table 5. (This is obtained by solving the formulation in Section 3, assuming parameter values are deterministic.) The same insight is observed for small $N = 1$ or 2 in Case 1, and $N = 1$ in Case 2.

The simulation-based hybrid heuristic can obtain feasible capacity values much easier (with half the total running time of the mixed integer program) and allows variation in different parameters in examining their impact. The capacity values obtained are more stable and lower than the deterministic model as problem size grows.

The Type 2 experiments are run under the scenario of moderate variability in time-based data ($c_a = c_T = c_\gamma = 1$). The relationship curves of total capacity versus N at different values of R and f_R are plotted for the two dynamic dispatch algorithms FCFS (Figure 5) and NN (Figure 6), respectively. These results can provide information to the following.

(i) Identify appropriate response time service level and allocate demand sites into service regions, given the number of facilities and total capacity available. Conversely, capacity can be estimated for a defined service region under a given response time service level.

(ii) Improve service level with a given capacity. For instance in the lowest requirement curve ($R = 90$ min. and $f_R = 50\%$), point A in Figure 5 represents a single facility and 16 service units. It can be improved to service level ($R = 60$ min., $f_R = 50\%$) by setting up

Table 5: Impact of doubling travel time versus doubling demand request on total capacity in static example (Figures 1 and 2).

Case	Number of facilities (N)	Minimum capacity (number of service units)		
		Original (Appendix A)	Actual demand request doubled (4 requests per node)	Actual travel time doubled (30 min. between every two nodes)
1	1	4	7	(Infeasible)
	2	2	4	8
	4	4	4	4
2	1	3	3	4
	2	2	3	3
	4	4	4	4

Case 1: (worst case) all requests arrive at time 0.
Case 2: (average case) half of the requests arrive at time 0, the other half at the 30th minute in an hour.

(at least 1) additional facility. Similarly in a higher requirement curve (R = 90 min., f_R = 90%), point B with 22 service units in a single facility can be improved to service level (R = 60 min., f_R = 90%) by setting up (at least 1) additional facility and with fewer units. Point C in Figure 6 illustrates a similar characteristic with another dynamic dispatch algorithm (NN). These improvements are possible typically when only a few facilities are set up and service level is not high.

(iii) Adopt an appropriate dispatching rule. When the response time service level is more demanding (small R and/or high f_R), the dispatch algorithm FCFS results in lower total capacity than NN since the order of service in FCFS is approximately following the call arrival order. On the other hand, NN will be beneficial when the response time requirement is moderate (e.g., R = 90 min., f_R = 50%), as there is more flexibility for a server to complete nearby available requests, thereby increasing productivity per server.

Additional experiments are conducted for requirement beyond the highest service level (R = 30 min, f_R = 90%) in Figures 5 and 6. Figure 7 represents the service level curves (R = 30 min, f_R = 95%) for the two dispatch algorithms. No capacity is feasible for the smallest N (= 4) when considering uncertainties, even if it is feasible under average parameter values. They tend to have large fluctuation in capacity compared to lower service level curves. However, they can still reflect the initial trend of reduction in total capacity for decentralized facilities. As N increases, the capacity stabilizes for some range of N but will increase again with larger N. In general, decentralization results in shorter travel distances to the next assigned request, quicker response time per trip at the expense of more service regions and facilities to be administered. Here, it is shown that when sequential trip travel is allowed for servicing requests, the minimum total capacity does not decrease monotonically with increase in number of facilities, but capacity will increase beyond a certain limit of N. Figures 5 and 6 also show that the higher the service level requirement (small R and large f_R), the wider the separation from the lower service level curves. Besides in the additional experiments, more fluctuation in total capacity is observed as time-based parameters increase its variation (larger coefficient of variation).

The impact of sharing capacity (Section 4.3) can allow some demand nodes to be "covered" by more than one facility. This occurs more frequently in nodes with small demand intensities. Hence, in the dynamic environment, such nodes could have second (or multiple) coverage provided by another facility (or service region) when service units in the first responsible facility are busy. The savings in total capacity are not much in the experiments, at most one or two units. Other capacity sharing models could be explored in sequential trip travel. This would be similar to split delivery options adopted in some vehicle routing operations.

6. Sensitivity of Results to Model Assumptions and Methodology Proposed

After solving the problem based on assumptions (1)–(10) in Section 3, a natural step is to realize certain model assumptions may limit the solution quality or oversimplify reality, such as the following:

(2) each demand site is to be assigned to exactly one facility;

(4) all requests are treated with the same priority.

Relaxing assumption (2) implies allowing service units from one or more facilities to respond to a request. Accordingly, the constraints related to assignment variables (y_{ij}) in the mathematical formulation (Section 3) could be modified. However, a larger service region may imply larger travelling time and response time which may offset the benefits from capacity sharing. This should be further investigated. Assumption (4) is unrealistic for emergency response systems. One approach is to apply a two-priority class to differentiate between requests and adopt different strategies, for example, response time service levels, dispatch methods, for each class, which is simply an extended version of the current problem.

Simulation is only one approach to model uncertainty when analytical formulas are unavailable. It requires some information on the distribution of uncertain parameters and lower/upper bounds are necessary to avoid using extreme simulated values. Different distribution assumed for an uncertain parameter may result in different total capacity values. However, the general trend of the curve relating total capacity and number of facilities is expected to be much the same. When the response time service level requirement is high (small R and/or large f_R), the curve tends to show large fluctuation in capacity, possibly due to the randomness nature of simulation. Despite requiring longer computational time and more coding effort, the advantages of applying simulation to this research include the following.

(i) Revealing an infeasible problem with uncertain parameters even if the problem seems feasible based on average parameter values. When setting up only a few facilities, even if the average direct travel time between facilities and assigned nodes is within the average response time limit, simulation can reveal the solution is infeasible when allowing variability in parameters (e.g., Figure 7 when $N = 4$, static 4-node example in Figure 1, and Table 5 when $N = 1$, and actual travel time is twice the average travel time).

(ii) Identifying the significant uncertain parameter(s) with larger impact on the results. When only a few facilities are set up, the variability in travel time tends to increase capacity more than variability in demand in achieving the same response time service level (e.g., Tables 1–4 when N is the smallest number).

The ejection and rejection procedure is one of the iterative improvement approaches. Other global or local search algorithms could be used as alternative, together with memory of feasible/infeasible regions. Performance comparison could be made first in the deterministic problems (e.g., Section 3 formulation assuming all deterministic parameters; mixed integer program in Appendix C) before considering the uncertainties.

The two dynamic dispatch methods could be improved by dynamic/adaptive scheduling algorithms which consider multiple server-request assignments simultaneously. The advantage of using simple (well-known) dispatch methods here is to avoid increasing the running time when the iterative simulation approach is adopted for this three-level integrated problem.

7. Conclusions

This study has provided an analytical framework to tackle a three-level territory planning problem involving simultaneous decisions on facility location (first level), demand allocation (second level), and resource capacity (third level) under uncertain demand, travel time, and service time when sequential trip travel is also allowed. Response time performance requirement is common in many industries. It may appear in the form of an upper limit (R) imposed on the average response time or a minimum percentage (f_R) of served requests satisfying a response time upper limit (R). In emergency systems planning, Carter et al. [44] showed that defining response areas for each ambulance will decrease the average response time. Bandara et al. [39] proposed to incorporate better dispatching rule for ambulances together with defined response areas by ambulance in their future research. Without considering call priorities and operational constraints, the three-level integrated problem is related in a way by considering the ambulance location as a facility, response areas as allocated demand sites, and number of ambulances (at each location) as resource capacity.

A simulation-based hybrid heuristic offers a flexible approach to test uncertainty factors and different servicing strategies. The current framework for a single period can be extended to a multiperiod planning problem by increasing the number of replications (with period-specific parameters) in the simulation model (SIM) to determine the capacity decision by period when the high-level decisions (facility location and demand allocation) are given or temporarily fixed. The relationship of this three-level problem with the deterministic bounded latency problem (BLP) is pointed out. When all parameters are deterministic and with unit request per node, the three-level problem corresponds to a BLP with location decisions and the latency bound given by the response time limit R. This can be formulated as a mixed integer program (MIP). Both the simulation-based heuristic and MIP are tested on twelve travelling salesman problem (TSP) data sets containing 29 to 262 nodes [40], assuming unit request per node. The heuristic is run with three different factors (interarrival time distribution, travel time variability, and dynamic dispatch algorithm) where each factor has two alternatives, resulting in eight versions. For small problems of up to 100 nodes, the heuristic produces solutions more efficiently and capacity values are comparable with the MIP results. For larger problems, the heuristic results are much lower than the best MIP results even with half the running time. The second type of experiments simulates the delivery service environment with local data and operating parameters from two local sources and a third source in Taiwan. Arrivals of requests are simulated. Results show that a policy with few facilities can achieve higher response time service level with the same or lower total capacity by operating with more facilities up to a certain limit. Beyond this limit, total capacity will increase. Some future research directions include the three-level problem with heterogeneous resources and servicing strategies, like capacity sharing and dynamic/adaptive scheduling of real-time requests.

Appendices

A. Static Example of 4 Nodes

Demand arrival rate per node = $\lambda = 2$ per hour; travel time between every pair of nodes = $t = 15$ min.; on-site processing time = $\gamma = 5$ min.

Response time requirement: average response time $\leq R = 15$ min.

Let F_i denote node i if a facility is set up, N_i otherwise ($i = 1, 2, 3, 4$).

Let i_k denote the kth request from node i ($i = 1, 2, 3, 4$; $k = 1, 2$).

For more details see Tables 6 and 7.

B. A Set-Covering Model with an Additional Constraint to Explore Sharing of Capacity Across Regions

Parameters. Consider the following:

s_j = capacity of service region $j \in P_F$,

D_i = set of service regions (j) in P_F that includes node $i = 1, \ldots, m$.

Decision Variables. Consider the following:

x_j = 1 if service region j is selected, 0 otherwise, $j \in P_F$.

Set-Covering Model with an Additional Constraint. Consider

$$\text{Minimize } Z = \sum_{j \in P_F} s_j \cdot x_j \tag{B.1}$$

subject to

$$\sum_{j \in D_i} x_j \geq 1, \quad i = 1, \ldots, m \tag{B.2}$$

$$\sum_{j \in P_F} x_j = N \tag{B.3}$$

$$x_j = 0, 1, \quad j \in P_F. \tag{B.4}$$

The objective function in constraint (B.1) minimizes the total capacity by selecting an optimal set of feasible regions in P_F. Constraint (B.2) requires each demand node in the territory to be included in at least one service region. Constraint (B.3) is the additional constraint requiring a total of N facilities to be selected (assumption (5) in Section 3). The selection decision variables of service regions are declared in constraint (B.4).

C. The Bounded Latency Problem (BLP) with Location Decisions

Parameters. Consider the following:

m = number of facility and demand nodes in network,

N = number of facilities to be selected,

s = (artificial) source node of network,

e = (artificial) sink node of network,

A = set of arcs in network = $\{(s, i), (i, e), (h, i) \mid h, i = 1, \ldots, m\}$,

A' = set of arcs in A excluding those linked to source, sink, or between a node and itself = $\{(h, i) \in A \mid h \neq i, s, i \neq e\}$,

t_{hi} = average travel time from node h to node i, $h, i = 1, \ldots, m$ (assume $t_{ii} = t_{si} = t_{ie} = 0$),

γ_h = on-site processing time of node $h = 1, \ldots, m$.

Decision Variables. Consider the following:

x_j = 1 if a facility is set up at node j, 0 otherwise, $j = 1, \ldots, m$,

$f_{h,i}$ = flow on arc $(h, i) \in A$,

$a_{h,i}$ = 1 if arc (h, i) is used, 0 otherwise, $(h, i) \in A'$,

τ_i = server arrival time at node i (used in the subtour elimination constraints), $i = 1, \ldots, m$.

Mixed Integer Program. Consider

$$\text{Minimize } Z = \sum_{(s,i) \in A} f_{s,i} \tag{C.1}$$

subject to

$$N \leq \sum_{(s,i) \in A} f_{s,i} \leq m - N \tag{C.2}$$

$$\sum_{j=1}^{m} x_j = N \tag{C.3}$$

$$f_{s,i} \leq m \cdot x_i, \quad i = 1, \ldots, m \tag{C.4}$$

$$f_{h,i} + x_h + x_i \leq 2, \quad (h, i) \in A' \tag{C.5}$$

$$f_{h,i} + x_i \leq 1, \quad (h, i) \in A' \tag{C.6}$$

$$\sum_{(h,i) \in \{A \mid h \neq i\}} f_{h,i} = f_{i,i}, \quad i = 1, \ldots, m \tag{C.7}$$

$$f_{i,i} = \sum_{(i,h) \in \{A \mid i \neq h\}} f_{i,h}, \quad i = 1, \ldots, m \tag{C.8}$$

$$1 \leq f_{i,i} \leq (m - 1) \cdot x_i + 1, \quad i = 1, \ldots, m \tag{C.9}$$

$$\tau_i \leq R, \quad i = 1, \ldots, m \tag{C.10}$$

$$a_{h,i} \leq f_{h,i} \leq m \cdot a_{h,i}, \quad (h, i) \in A' \tag{C.11}$$

$$\tau_h + (R + \gamma_h + t_{hi}) \cdot a_{h,i} - \tau_i \leq R, \quad (h, i) \in A' \tag{C.12}$$

$$\begin{gathered} f_{h,i} \geq 0, \quad (h, i) \in A, \\ a_{h,i} = 0, 1, \quad (h, i) \in A', \\ x_i = 0, 1, \quad \tau_i \geq 0, \quad i = 1, \ldots, m. \end{gathered} \tag{C.13}$$

The main difference with the BLP is the additional location decisions of N facilities and its assigned demand nodes. By network construction, the source node (s) is linked to every facility node which could be set up at any facility/demand node, without loss of generality. The allocation decisions (of demand nodes) are represented by nodes visited on the path(s) originating from a facility node. In every feasible solution, each demand node will have exactly

Table 6: Case 1 (worst case) with all requests arriving at time 0.

First level decision Number of facilities {facility nodes}	Second level decision Facility node: assigned node(s)	Third level decision Minimum capacity (number of units)	Assigned facility node	Service unit# Route: request# (earliest arrival time)	Sum of response time (min.) to requests	Average response time per request ($\leq R = 15$)	Average travel time per request (min.)
4 $\{F_1, F_2, F_3, F_4\}$	F_i: F_i ($i = 1, 2, 3, 4$)	4	Unit i: F_i ($i = 1, 2, 3, 4$)	Unit i: i_1 (0) → i_2 (5) ($i = 1, 2, 3, 4$)	Unit i: 0 + 5 = 5 ($i = 1, 2, \ldots, 4$)	2.5	0
2 $\{F_1, F_3\}$	F_1: F_1, N_2 F_3: F_3, N_4	2	Unit 1: F_1 Unit 2: F_3	Unit 1: 1_1 (0) → 1_2 (5) → 2_1 (25) → 2_2 (30) Unit 2: 3_1 (0) → 3_2 (5) → 4_1 (25) → 4_2 (30)	Unit 1: 0 + 5 + 25 + 30 = 60 Unit 2: 0 + 5 + 25 + 30 = 60	15	3.75
1 $\{F_1\}$	F_1: (all nodes)	4	Unit i: F_1 ($i = 1, 2, 3, 4$)	Unit 1: 1_1 (0) → 1_2 (5) Unit 2: F_1 → 2_1 (15) → 2_2 (20) Unit 3: F_1 → 3_1 (15) → 3_2 (20) Unit 4: F_1 → 4_1 (15) → 4_2 (20)	Unit 1: 0 + 5 Unit i: 15 + 20 = 35 ($i = 2, 3, 4$)	13.75	5.625

Table 7: Case 2 (average case) with requests arriving at time 0 and the 30th min. in an hour.

First level decision Number of facilities {facility nodes}	Second level decision Facility node: assigned node(s)	Third level decision Minimum capacity (number of units)	Assigned facility node	Service unit# Route: request# (earliest arrival time)	Sum of response time (min.) to requests	Average response time per request ($\leq R = 15$)	Average travel time per request (min.)
4 $\{F_1, F_2, F_3, F_4\}$	F_i: F_i ($i = 1, 2, 3, 4$)	4	Unit i: F_i ($i = 1, 2, 3, 4$)	Unit i: i_1 (0) → i_2 (5) ($i = 1, 2, 3, 4$)	Unit i: 0 + 0 = 0 ($i = 1, 2, \ldots, 4$)	0	0
2 $\{F_1, F_3\}$	F_1: F_1, N_2	2	Unit 1: F_1	Unit 1: 1_1 (0) → 2_1 (20) → 2_2 (25) → 1_2 (50)	Unit 1: 0 + 20 + 0 + 20 = 40	10	7.5
	F_3: F_3, N_4		Unit 2: F_3	Unit 2: 3_1 (0) → 4_1 (20) → 4_2 (25) → 3_2 (50)	Unit 2: 0 + 20 + 0 + 20 = 40		
1 $\{F_1\}$	F_1: (all nodes)	3	Unit i: F_1 ($i = 1, 2, 3$)	Unit 1: 1_1 (0) → 2_1 (20) → 2_2 (25) → 1_2 (50) Unit 2: F_1 → 2_1 (15) → 2_2 (20) Unit 3: F_1 → 3_1 (15) → 3_2 (20)	Unit 1: 0 + 20 + 0 + 20 = 40 Unit 2: 15 + 0 = 15 Unit 3: 15 + 0 = 15	8.75	7.5

one facility node as its predecessor. Unit request is assumed for each facility/demand node and all requests are available at time 0. The response time requirement (R), represented by constraints (C.10), is imposed on the server arrival time at every node (except sink node).

The flow variables ($f_{h,i}$) represent the number of service units travelling between two nodes (from h to i). The objective in (C.1) determines the minimum capacity (number of service units) to fulfill the response time requirement (R) for all requests. This capacity is represented by the total outflow from the source node (s) and its natural lower and upper bounds are given in constraint (C.2). The selection of N facilities (Section 3, assumption (5)) and their locations are expressed in equation (C.3). Constraint (C.4) restricts that a facility node (i) can receive at most m units (= total number of requests, one from each facility/demand node) of flow from the source node (s). The inflow to the node (i) represents the number of requests that can be served through this facility. Constraint (C.5) formulates the independence condition of facilities that no flow is allowed between two facility nodes. Constraint (C.6) restricts that a facility node (i) should not receive flow from a demand node (h). Flow balance at each node (except source and sink nodes) is formulated by constraints (C.7) and (C.8). Arc (i, i) in these two constraints enforces at least one service unit to visit node i $(= 1, \ldots, m)$ by imposing a lower bound constraint (C.9). If node i is a facility, the upper bound on arc flow is m $(= m - 1 + 1)$, the sum of requests from all nodes, and 1 otherwise. The response time requirement (latency bound or waiting time) is formulated by constraint (C.10). The relationship between the flow on an arc and the use of the arc is formulated by constraint (C.11) which enables the formulation of subtour elimination constraints in (C.12). The last constraint (C.13) declares the type of decision variables and their restrictions.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The work described in this paper was fully supported by the Strategic Research Grant from City University of Hong Kong (Project no. 7002524).

References

[1] R. Aboolian, Y. Sun, and G. J. Koehler, "A location-allocation problem for a web services provider in a competitive market," *European Journal of Operational Research*, vol. 194, no. 1, pp. 64–77, 2009.

[2] A. B. Arabani and R. Z. Farahani, "Facility location dynamics: an overview of classifications and applications," *Computers & Industrial Engineering*, vol. 62, no. 1, pp. 408–420, 2012.

[3] R. Z. Farahani and M. Hekmatfar, *Facility Location: Concepts, Models, Algorithms and Case Studies*, Physica, Heidelberg, Germany, 2009.

[4] Z. Azarmand and E. N. Jamie, "Location allocation problem," in *Facility Location: Concepts, Models, Algorithms and Case Studies*, R. Z. Farahani and M. Hekmatfar, Eds., pp. 93–109, Physica, Heidelberg, Germany, 2009.

[5] R. Aboolian, O. Berman, and Z. Drezner, "Location and allocation of service units on a congested network," *IIE Transactions*, vol. 40, no. 4, pp. 422–433, 2008.

[6] S. S. Syam, "A multiple server location-allocation model for service system design," *Computers & Operations Research*, vol. 35, no. 7, pp. 2248–2265, 2008.

[7] M. Turkensteen and A. Klose, "Demand dispersion and logistics costs in one-to-many distribution systems," *European Journal of Operational Research*, vol. 223, no. 2, pp. 499–507, 2012.

[8] T. G. Crainic, G. Perboli, S. Mancini, and R. Tadei, "Two-echelon vehicle routing problem: a satellite location analysis," *Procedia—Social and Behavioral Sciences*, vol. 2, no. 3, pp. 5944–5955, 2010.

[9] T. G. Crainic, N. Ricciardi, and G. Storchi, "Models for evaluating and planning city logistics systems," *Transportation Science*, vol. 43, no. 4, pp. 432–454, 2009.

[10] J. Brimberg, A. Mehrez, and G. O. Wesolowsky, "Allocation of queuing facilities using a minimax criterion," *Location Science*, vol. 5, no. 2, pp. 89–101, 1997.

[11] O. Kariv and S. L. Hakimi, "An algorithmic approach to network location problems II: the p-medians," *SIAM Journal on Applied Mathematics*, vol. 37, no. 3, pp. 539–560, 1979.

[12] S. Sahni and T. Gonzalez, "P-complete approximation problems," *Journal of the Association for Computing Machinery*, vol. 23, no. 3, pp. 555–565, 1976.

[13] R. Jothi and B. Raghavachari, "Approximating the k-traveling repairman problem with repairtimes," *Journal of Discrete Algorithms*, vol. 5, no. 2, pp. 293–303, 2007.

[14] I. Averbakh and O. Berman, "Routing and location-routing p-delivery men problems on a path," *Transportation Science*, vol. 28, no. 2, pp. 162–166, 1994.

[15] M. Jamil, R. Batta, and D. M. Malon, "The traveling repairperson home base location problem," *Transportation Science*, vol. 28, no. 2, pp. 150–161, 1994.

[16] A. V. Hill, S. T. March, C. J. Nachtsheim, and M. S. Shanker, "An approximate model for field service territory planning," *IIE Transactions*, vol. 24, no. 1, pp. 2–10, 1992.

[17] S. C. K. Chu and C. K. Y. Lin, "Manpower allocation model of job specialization," *Journal of the Operational Research Society*, vol. 44, no. 10, pp. 983–989, 1993.

[18] Q. Tang, G. R. Wilson, and E. Perevalov, "An approximation manpower planning model for after-sales field service support," *Computers & Operations Research*, vol. 35, no. 11, pp. 3479–3488, 2008.

[19] N. Geroliminis, K. Kepaptsoglou, and M. G. Karlaftis, "A hybrid hypercube—genetic algorithm approach for deploying many emergency response mobile units in an urban network," *European Journal of Operational Research*, vol. 210, no. 2, pp. 287–300, 2011.

[20] R. T. Wong, "Vehicle routing for small package delivery and pickup services," in *The Vehicle Routing Problem: Latest Advances and New Challenges*, B. L. Golden, S. Raghavan, and E. A. Wasil, Eds., pp. 475–485, Springer, New York, NY, USA, 2008.

[21] S. Powell, "Using linear programming to simulate service engineers," *Journal of the Operational Research Society*, vol. 50, no. 12, pp. 1252–1255, 1999.

[22] P. Kolesar, W. Walker, and J. Hausner, "Determining the relation between fire engine travel times and travel distances in New Yor City companies," *Operations Research*, vol. 23, no. 4, pp. 614–627, 1975.

[23] E. K. Lee, C. H. Chen, F. Pietz, and B. Benecke, "Modeling and optimizing the public-health infrastructure for emergency response," *Interfaces*, vol. 39, no. 5, pp. 476–490, 2009.

[24] E. K. Lee, F. Piez, and B. Benecke, "Service networks for public health and medical preparedness: medical countermeasures dispensing and large-scale disaster relief efforts," in *Handbook of Operations Research for Homeland Security*, J. W. Herrmann, Ed., vol. 183 of *International Series in Operations Research & Management Science*, pp. 167–196, 2013.

[25] R. C. Larson, "A hypercube queuing model for facility location and redistricting in urban emergency services," *Computers & Operations Research*, vol. 1, no. 1, pp. 67–95, 1974.

[26] R. C. Larson and A. R. Odoni, *Urban Operations Research*, Prentice Hall, Englewood Cliffs, NJ, USA, 1981, http://web.mit.edu/urban_or_book/www/book/.

[27] S. Lee, "The role of centrality in ambulance dispatching," *Decision Support Systems*, vol. 54, no. 1, pp. 282–291, 2012.

[28] P. Beraldi and M. E. Bruni, "A probabilistic model applied to emergency service vehicle location," *European Journal of Operational Research*, vol. 196, no. 1, pp. 323–331, 2009.

[29] H. Zhong, R. W. Hall, and M. Dessouky, "Territory planning and vehicle dispatching with driver learning," *Transportation Science*, vol. 41, no. 1, pp. 74–89, 2007.

[30] L. M. Wein, "Neither snow, nor rain, nor anthrax," *The New York Times*, 2008, http://www.nytimes.com/2008/10/13/opinion/13wein.html?_r=1.

[31] A. Richter and S. Khan, "Pilot model: judging alternate modes of dispensing prophylaxis in Los Angeles County," *Interfaces*, vol. 39, no. 3, pp. 228–240, 2009.

[32] N. Coskun and R. Erol, "An optimization model for locating and sizing emergency medical service stations," *Journal of Medical Systems*, vol. 34, no. 1, pp. 43–49, 2010.

[33] M. B. Pedersen, T. G. Crainic, and O. B. G. Madsen, "Models and tabu search metaheuristics for service network design with asset-balance requirements," *Transportation Science*, vol. 43, no. 2, pp. 158–177, 2009.

[34] A. Hoff, A. Lium, A. Løkketangen, and T. G. Crainic, "A metaheuristic for stochastic service network design," *Journal of Heuristics*, vol. 16, no. 5, pp. 653–679, 2010.

[35] X. Li and A. G.-O. Yeh, "Integration of genetic algorithms and GIS for optimal location search," *International Journal of Geographical Information Science*, vol. 19, no. 5, pp. 581–601, 2005.

[36] J.-R. Lin, S. Yan, and C. W. Lai, "International express courier routing and scheduling under uncertain demands," *Engineering Optimization*, vol. 45, no. 7, pp. 881–897, 2013.

[37] Annual Transport Digest. Transport Department. Hong Kong, 2012, http://www.td.gov.hk/mini_site/atd/2012/eng/section5/section5_18.html.

[38] A. Mandle, A. Jaiswal, B. Dod, and R. Lokhande, "Taxi automation using real time adaptive scheduling," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, no. 3, pp. 592–594, 2014.

[39] D. Bandara, M. E. Mayorga, and L. A. McLay, "Priority dispatching strategies for EMS systems," *Journal of the Operational Research Society*, vol. 65, no. 4, pp. 572–587, 2014.

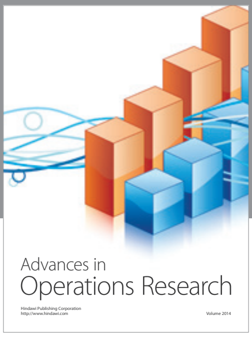
[40] TSPLIB, Ruprecht-Karls-Universität, Heidelberg, Germany, http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/.

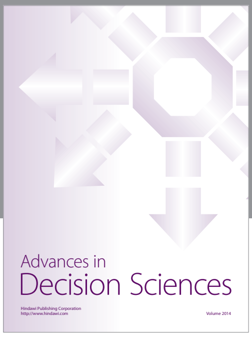
[41] C. K. Y. Lin and R. C. W. Kwok, "Multi-objective metaheuristics for a location-routing problem with multiple use of vehicles on real data and simulated data," *European Journal of Operational Research*, vol. 175, no. 3, pp. 1833–1849, 2006.

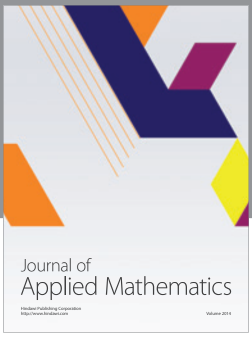
[42] "Delivery_data," City University of Hong Kong, Hong Kong, http://personal.cityu.edu.hk/~mslincky/Demand_distances.xls.

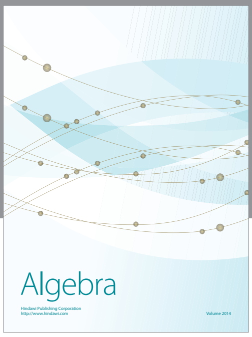
[43] C. K. Y. Lin, "A vehicle routing problem with pickup and delivery time windows, and coordination of transportable resources," *Computers & Operations Research*, vol. 38, no. 11, pp. 1596–1609, 2011.

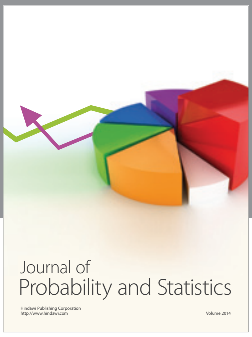
[44] G. M. Carter, J. M. Chaiken, and E. Ignall, "Response areas for two emergency units," *Operations Research*, vol. 20, no. 3, pp. 571–594, 1972.


Advances in
Operations Research

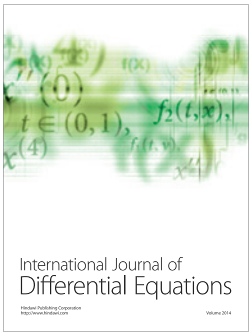

Advances in
Decision Sciences


Journal of
Applied Mathematics

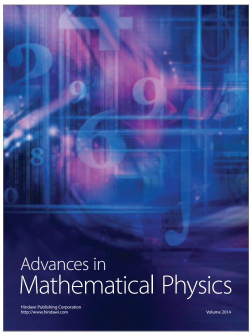

Algebra

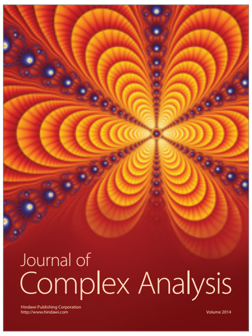

Journal of
Probability and Statistics

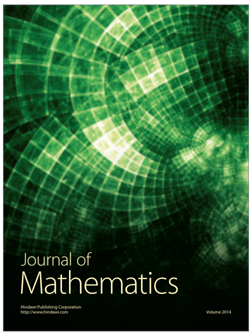
The Scientific
World Journal

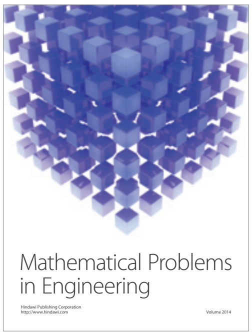

International Journal of
Differential Equations

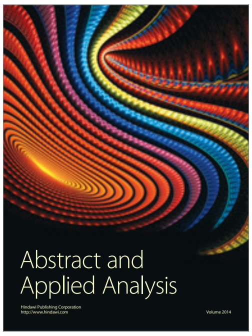
International Journal of
Combinatorics

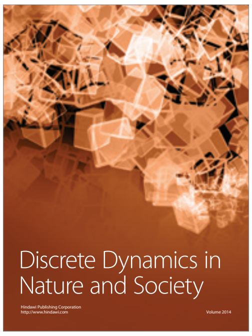

Advances in
Mathematical Physics

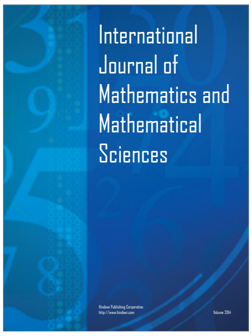

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

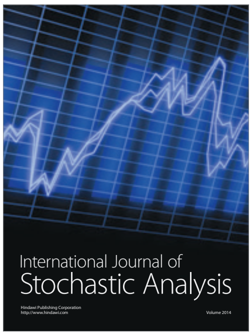

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization