Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

SELECTIVE PICK-UP AND DELIVERY PROBLEM: A SIMHEURISTIC APPROACH

Tejas Ghorpade

Industrial Engineering and Operations Research
Indian Institute of Technology, Bombay
Powai, Mumbai 411027, INDIA

Canan G. Corlu

Metropolitan College
Boston University
Boston, MA 02215, USA

ABSTRACT

The one-commodity pick-up and delivery traveling salesman problem (1-PDTSP) concerns the transportation of single-type goods that are picked up from supply locations to be delivered to the demand points while minimizing the transportation cost. A variant of the 1-PDTSP is the selective pick-up and delivery problem (SPDP), which relaxes the requirement that all pick-up locations need to be visited. The SPDP is applicable in several areas including food redistribution operations, where excess edible foods from restaurants and food vendors are collected and delivered to food banks or meal centers, where they can be made available to those in need. Because the SPDP is an NP-hard problem, metaheuristic algorithms have been proposed in the literature to solve it. However, these algorithms make the assumption that all inputs are deterministic, which might not be the case in practice. This paper considers a stochastic SPDP and proposes a simheuristic algorithm that integrates a GRASP metaheuristic with Monte Carlo simulation.

1 INTRODUCTION

The pick-up and delivery problem (PDP) has been studied extensively in the literature due to its application in several areas, including – but not limited to – dial-a-ride systems, reverse logistics, bike repositioning operations, and food redistribution operations. Several variants of single-commodity PDPs have been introduced in the literature (see Berbeglia et al. 2007 for a review). One of the well-known variants is the one-commodity pick-up and delivery traveling salesman problem (1-PDTSP), which was first introduced by Hernández-Pérez and Salazar-González (2004). 1-PDTSP concerns the transportation of a single commodity among a set of pick-up nodes (suppliers for the commodity) and delivery nodes (demand points). The goal of the 1-PDTSP (or typical single commodity PDP) is to distribute commodities between pick-up and delivery nodes via a single vehicle, so as to minimize the total transportation cost.

A novel variant of the 1-PDTSP is referred to as the selective pick-up and delivery problem (SPDP). It has been introduced by Ting and Liao (2013). This variant relaxes the typical assumption in the 1-PDSTP literature that all pick-up and delivery nodes need to be visited by allowing some of the pick-up nodes to be skipped. In other words, it is sufficient to visit a subset of all pick-up nodes that will enable the satisfaction of the demand in the delivery nodes. This approach allows for reducing the transportation cost while still satisfying the demands of the delivery nodes. This arises as a practical situation in the bike-repositioning problem where some rental bikes need to be transported between stations by a truck. The truck picks up the bikes from several stations and distributes them to other stations where there are demand and reservations. In this case, the truck does not need to visit all the stations – it is, in fact, sufficient to visit *some* of the stations that will enable the satisfaction of requests at the more demanding stations. Another important application, which in fact motivates this paper, is the food redistribution problem. The food rescue and distribution system involves collecting excess edible food from restaurants, dining facilities, grocery markets, food vendors, and farmers markets and delivering it to agencies or meal centers where it can be made available to those in need. Certain non-profit organizations in many countries perform this task of collecting the excess food and distributing it to the locations where it is required (Gunes et al. 2010).

From an optimization point of view, Ting and Liao (2013) show that the SPDP is an NP-hard problem. The authors propose a memetic algorithm that is based on a genetic algorithm and local search. This algorithm is shown to improve upon the genetic algorithm and tabu search in terms of solution quality and computational speed. Ho and Szeto (2016) propose a GRASP algorithm with path relinking for SPDP. The algorithm is shown to perform better than the memetic algorithm proposed in Ting and Liao (2013) by providing 5.72 % improvement in the existing solution. The limited existing literature on SPDP makes the assumption that the demand at the delivery nodes and supply at the pick-up nodes are deterministic. However, this may not be the case in practice. Specifically, in the food redistribution problem, there could be fluctuations in the amount of food donated from the supply locations and the demand may also vary in certain locations. In this case, routes planned for a particular scenario for the deterministic case may not provide an efficient solution for the stochastic scenario. *To the best of our knowledge, this is the first paper that studies a stochastic SPDP and proposes a simheuristic algorithm to solve it. Furthermore, in the traditional SPDP literature, the main assumption is that total supply is always greater than the total demand. We then relax this assumption and look at the cases where there is not enough supply to fulfill the total demand, which is a reality in the food redistribution problem. This leads to a new mixed integer linear model that we propose in this paper.*

Simheuristics are a simulation-optimization technique, which combines metaheuristics with simulation for handling stochastic inputs in combinatorial optimization problems. It has been used extensively in the solution of NP-hard combinatorial optimization problems such as vehicle-routing problems and inventory-routing problems with stochastic components (see, e.g., Gonzalez-Martin et al. 2014, Guimarans et al. 2018, and Reyes-Rubiano et al. 2019). Pérez et al. (2015) provide a review of simheuristics as a simulation-optimization approach to solve stochastic combinatorial optimization problems. Recently, Juan et al. (2018) review the applications of simheuristics, specifically in the area of logistics and transportation. A close look at the existing literature reveals that simheuristics have not yet been employed to solve any variant of the PDP. In this work, we contribute to the literature by proposing a simheuristic algorithm to solve the selective pick-up and delivery problems with stochastic demands. Specifically, we have modified the GRASP heuristic given by Ho and Szeto (2016) and combined this metaheuristic with Monte Carlo simulation to generate a simheurisitic algorithm. A GRASP heuristic has been integrated into a simheuristic framework by Maccarrone et al. (2018) in order to solve the integrated resource allocation and scheduling problem. A more related application is the simGRASP algorithm proposed by Festa et al. (2018) to solve the vehicle routing problem with stochastic demands.

We organize the remainder of this paper as follows: Section 2 provides a mathematical formulation for the SPDP. Section 3 reviews the GRASP algorithm used to solve the deterministic SPDP and proposes a simheuristic algorithm to solve the stochastic SPDP. Section 4 presents the computational experiments, and Section 5 concludes with future research directions.

2 MATHEMATICAL FORMULATION

2.1 Problem Description

We consider the problem with a single vehicle that must traverse pick-up and delivery nodes over a network. Let $G = (N, A)$ be the network where N is the set of nodes including the depot 0 and both the pick-up (P) and delivery (D) locations and A is the set of directed arcs representing the shortest path between any two given nodes. Because this is a single-commodity problem, each pick-up node $p \in P$ generates some positive supply $s_p > 0$ and each delivery node $d \in D$ has demand for certain quantities of the commodity that is represented as negative supply ($s_d < 0$). We assume that the depot does not have any supply or demand. A single *uncapacitated* vehicle is used to meet the maximum possible demands by transporting commodity from the supply nodes to the demand nodes such that it visits each delivery and pick-up node at most once. When the vehicle visits a supply node, the available quantity of goods is loaded into the vehicle, whereas when it arrives at a delivery location, it serves the maximum possible demand at that location. *Unlike the previous SPDP literature, this work does not assume that total supply is always greater than total demand.*

We further define N_P as the net supply over all supply nodes; i.e., $N_P = \sum_{p \in P} s_p$ and N_D as the net demand over all demand nodes; i.e., $N_D = \sum_{d \in D} s_d$. The tour T is a series of nodes visited by the vehicle starting from the depot and returning to the depot, represented by $t_0, t_1, t_2, ...t_n, t_{n+1}$ where t_0 and t_{n+1} represent the depot and intermediate nodes being either pick-up or delivery nodes, $\{t_1, t_2...t_n\} \in P \cup D$. A tour must be such that, when $N_P \geq N_D$ every delivery node is visited and the vehicle has sufficient inventory to meet demands at each node. If $N_P < N_D$, then the vehicle meets the maximum possible demand using the available supply.

This problem is illustrated by a small example in Figure 1, where both subfigures represent a single network with a single depot and nine locations. The set of pick-up locations is $P = \{1,2,3,5,7,8\}$ and the set of delivery nodes is $D = \{4,6,9\}$ with the corresponding deterministic supply values s_i represented in the two plots. The goal of this paper is to determine the vehicle tour T, starting from and returning to the depot such that total travel cost are minimized and maximum possible demand is satisfied when the demand and supply are stochastic. For simplicity, the optimal tours in Figure 1 assume deterministic supply and demand. In the plot on the left, the vehicle tour is determined as $T = \{0,1,2,4,5,6,8,9\}$. In this case, $N_P > N_D$ and the vehicle does not visit supply nodes 3 and 7 because demand can be satisfied without visiting these nodes. Similarly, in the plot on the right, the vehicle tour is represented as $T = \{0,1,2,3,4,5,7,8,9,0\}$. Here, $N_P < N_D$ and the demand node 6 is not visited because the available supply barely satisfies the demand at nodes 3 and 9.

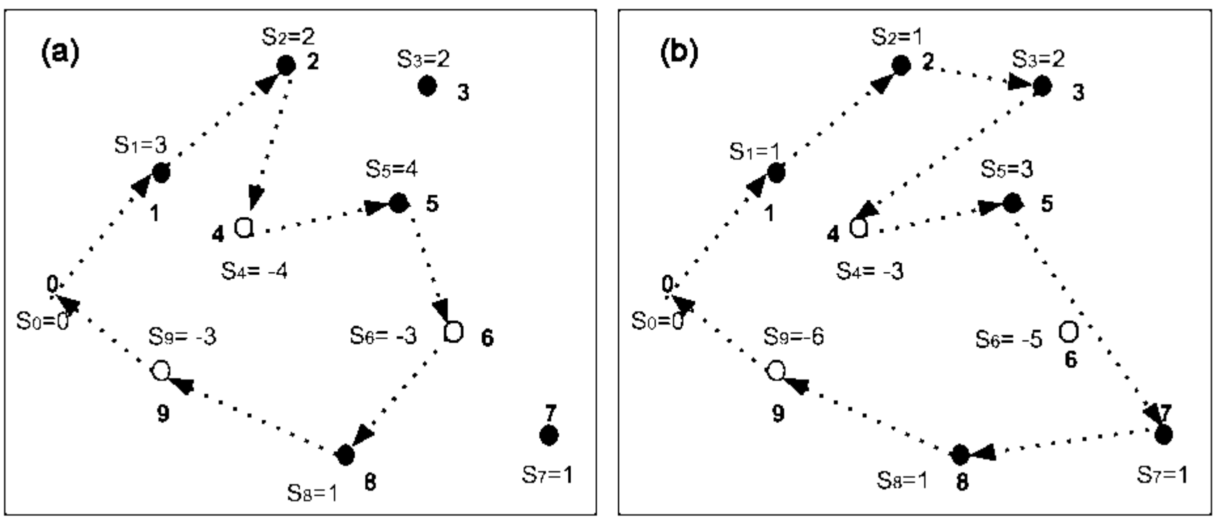

Figure 1: Tours for SPDP under alternate supply conditions, (a) net supply $>$ net demand, (b) net supply $<$ net demand.

2.2 Mathematical Model

This section presents a mixed linear integer program, whose solution provides the optimal solution for the deterministic values of supply and demand. Let N be a set of all the nodes in the network and A the set of directed arcs between them. These nodes consist of pick-up nodes P and delivery nodes D with supply value s_n at node $n \in N$ such that the value of $s_n \geq 0$ for $n \in S$ and $s_n < 0$ for $n \in D$. The capacity of the vehicle is given by Q. The cost associated with traversing the arc a_{ij} connecting nodes $i, j \in N$ are given by c_{ij}. The variable x_{ij} takes the value of 1 if the vehicle moves from node i to node j and it takes the value of 0, otherwise. l_i represent the load of the vehicle when it visits node $i \in N$.

$$\text{Maximize} \sum_{j\in D} (|s_j| \sum_{i\in N\setminus j} x_{ij}) - \sum_{ij\in A} c_{ij}x_{ij} \tag{1}$$

such that

$$\sum_{j\in N\setminus 0} x_{0j} = 1 \tag{2}$$

$$\sum_{j\in N\setminus 0} x_{j0} = 1 \tag{3}$$

$$\sum_{(ij)\in A} x_{ij} \leq 1 \qquad \forall i \in N\setminus 0 \tag{4}$$

$$\sum_{i:(ij)\in A} x_{ij} = \sum_{i:(ji)\in A} x_{ji} \qquad \forall j \in N\setminus 0 \tag{5}$$

$$l_0 = 0, u_0 = 0 \tag{6}$$

$$l_j \geq l_i + s_i - M(1 - x_{ij}) \qquad \forall i \in N,\ j \in N\setminus 0 \tag{7}$$

$$l_i \leq Q \sum_{j\in N} x_{ij} \qquad \forall i \in N \tag{8}$$

$$l_i \geq |s_i| \sum_{j\in N} x_{ij} \qquad \forall i \in D \tag{9}$$

$$x_{ij} \in \{0,1\} \qquad \forall i,\ j \in N \tag{10}$$

$$l_i = \mathbb{Z}^+ \qquad \forall i \in N \tag{11}$$

The objective (1) maximizes the total demand that is met while minimizing the cost of routing the vehicle. Thus, the maximum demand is satisfied irrespective of whether N_S is greater than or less than N_D. Constraints (2) and (3) ensure that each route starts and ends at the depot, whereas Constraint (4) restricts the vehicle to visit each node at most once. Equation (5) imposes flow balance on each node in the network. Constraint (6) sets the values of variables l and u to 0 at the depot. Constraint (7) computes the vehicle load at each node (M is a constant having a large value) and Constraint (8) ensures that the total load does not exceed the vehicle capacity at any location. Constraint (9) imposes that whenever the vehicle visits a demand node, there is sufficient supply to meet the demand at that location. This model does not allow for visiting a demand node for partial fulfillment of demand. Constraint (10) states that variable x_{ij} can only take binary values whereas Constraint (11) ensures that the vehicle load assumes non-negative integer values.

3 SOLUTION APPROACH

An optimal solution for the deterministic problem provided in Section 2 might give poor results when there is stochasticity in the supply and demand values. In this paper, we develop a simheuristic algorithm that combines Monte Carlo simulation with a metaheuristic algorithm to compute a solution that works well with stochastic inputs. It is important to note that the performance of the simheuristic algorithm depends on the quality of the deterministic algorithm used. Based on the detailed review of the existing literature on 1-PDTSP and SPDP, we conclude that the GRASP algorithm proposed by Ho and Szeto (2016) performs well in practical applications where each pick-up node does not need to be visited. Therefore, we build a simheuristic algorithm based on the randomized GRASP algorithm. The GRASP algorithm is described in Section 3.1 and we describe our simheuristic algorithm in Section 3.2.

3.1 Deterministic GRASP Algorithm

The GRASP metaheuristic is a multi-start algorithm based on a randomized construction process combined with local search to solve combinatorial optimization problems. In the first phase, a solution is built iteratively by randomly adding elements from a Restricted Candidate List (RCL). The RCL is generated by adding elements from the state space using a greedy function. This function measures the benefit or cost of including the element in the current solution. Once a feasible solution is constructed, the local search attempts to reach a local optimum with respect to a suitably defined neighbourhood structure. Randomly selecting element from the RCL ensures that different solutions are generated in the multi-start phase, which ultimately ensures that the solution is not trapped in local minima. At the end of the process, the best solution is returned as the final solution. A tour is deemed to be *feasible* if the load of the vehicle is greater than the demand at the node that is visited by the vehicle.

Ho and Szeto (2016) initially generate an optimal route over all the delivery nodes using the constructive phase and local search of GRASP followed by path-relinking. Finally, a feasible solution is obtained by minimum cost insertion of pick-up nodes in the constructed route. In this paper, we use a slightly modified version of this algorithm to solve the deterministic version of the SPDP. The modification is based on the idea that, even if we have an optimal route over the delivery locations, inserting pick-up routes may make the final route suboptimal. Ho and Szeto (2016) mention this as one of the limitations of their algorithm and state that it may not perform well when the number of pick-up nodes is large. Therefore, we first generate a randomized path over the delivery nodes using Algorithm 1 and then obtain a feasible path by inserting the pick-up nodes using Algorithm 2. These two steps constitute the construction phase of the GRASP heuristic.

Algorithm 1 is a constructive heuristic where a tour is constructed by sequentially inserting the delivery nodes. We start with a tour $T = (t_0, t_{n+1})$, where $t_0 = t_{n+1} = 0$ represent the depot, and create a copy of demand nodes $\bar{D}$. In the next step, we calculate the cost of inserting each demand node $d \in \bar{D}$ in T before the final visit to the depot; i.e., between t_n and t_{n+1}. A Restricted Candidate List is created. It includes all nodes where the cost are less than the threshold value $c(r) \leq c_{min} + \alpha(c_{max} - c_{min})$, which is determined by the parameter α and the minimum and maximum insertion cost c_{min} and c_{max}. An element $\hat{t}$ is randomly chosen from the RCL, inserted into T, and removed from $\bar{D}$. This is continued until all delivery nodes are inserted in the tour or until the sum of all demand over T exceeds the net supply. Thus, if including the demand node d in T causes the sum of demands over all $t \in T$ to exceed the available supply, this new node is not included in the tour. Thus, the algorithm does not consider partial fulfillment of demands at a given node.

To generate a feasible solution, supply nodes need to be added to the current tour T. The procedure for adding the pick-up nodes is given in Algorithm 2. In this algorithm, first the load l_k at each $k \in |T|$ is calculated as the sum of all positive and negative s_i values for all $i < k$. The position of the element in T with minimum negative load, y, is determined and the cost of inserting a pick-up node $r \in P$ at each position $o \in \{1, 2..y\}$ are calculated. The (r, o) pair with the least insertion cost is selected, r is inserted at position o in T, and the load values are updated. This procedure is repeated until there are no negative loads on the route or the entire supply is exhausted. Thus, we obtain a feasible route where the vehicle reaches each demand node with sufficient supply to satisfy the given demand.

This solution is further improved by a 2-opt local search with an included feasibility check. 2-opt is a classical neighbourhood operator where a neighbouring solution is obtained by destroying the original tour by removing non-adjacent arcs (i_u, i_{u+1}) and (i_v, i_{v+1}), and reconstructing the tour by adding arcs (i_u, i_v) and (i_{u+1}, i_{v+1}). If the cost of the new tour are less than the cost of the original tour and the feasibility condition (that the load of the vehicle when it reaches any demand node is positive) is satisfied, this new tour replaces the existing tour. This heuristic is run in a multi-start fashion and the route that gives the minimum cost C^G_{det} is selected in the final step.

We use this randomized deterministic algorithm to generate a simheuristic algorithm, which is described in the next section.

Algorithm 1: Greedy randomized path over delivery nodes.

Initialization: $T = (t_0, t_{n+1})$;
Set $\bar{D} = D$;
while $|\bar{D}| > 0$ **do**
  set $c_{min} = min_{r \in \bar{D}}\{c_{t_n,r} + c_{r,t_{n+1}} - c_{t_n,t_{n+1}}\}$
  set $c_{max} = max_{r \in \bar{D}}\{c_{t_n,r} + c_{r,t_{n+1}} - c_{t_n,t_{n+1}}\}$
  Set $RCL = \{r \in \bar{D} : c(r) \leq c_{min} + \alpha(c_{max} - c_{min})\}$
  Randomly select $\hat{t} \in RCL$
  if $\sum_{k \in T}(S_k) + S_i < \sum_{k \in P}(S_k)$ **then**
    Insert $\hat{t}$ between t_n and t_{n+1} in T
    Set $\bar{D} = \bar{D} \backslash \hat{t}$
  else
    break
  end
end

Algorithm 2: Adding pick-up nodes to the vehicle tour.

Initialization: T= Initial tour over delivery nodes;
$l_0 = 0$
for $k \in |T|$ **do**
  $l_k = l_{k-1} + S_{lk}$
end
$l_{min} = min_{k \in |T|}(l_k)$;
Set $\bar{P} = P$;
while $l_{min} < 0$ *or* $|\bar{P}| > 0$ **do**
  $y = min\{k \in |l| : l_k < 0\}$;
  (r, o) = $argmin_{r \in \bar{P}}, o = \{1, 2...y\}\{c_{t_{o-1},r} + c_{r,t_o} - c_{t_{o-1},t_o}\}$;
  Insert r in position o in T;
  Set $\bar{P} = \bar{P} \backslash r$;
  Update l and l_{min};
end

3.2 Simheuristic Approach

Figure 2 presents the flow of our simheuristic approach. We aim to determine a path for the SPDP such that the maximum demand can be served when both demand and supply are stochastic. The deterministic GRASP algorithm tries to minimize the total distance that the vehicle has to traverse while serving maximum demands when the supply and demands at each location are known. Thus, in the first stage, a randomized solution based on the GRASP heuristic is constructed as described in the previous section by taking the average demand and supply values as inputs. Let this solution be represented by x. Note that the proposed algorithm is a multi-start GRASP algorithm and generates a different solution every time the simulation is run.

In the next step, we aim to determine the stochastic cost associated with the given solution. Since simulation requires high computational time, this is done in two steps. Initially, a small number of simulation runs are performed and an elite set E of promising solutions is constructed. This is termed as *fast simulation*. In the later stage, *extended simulation* runs are performed, where the members of the elite set undergo a large number of simulation runs, and the solution with the lowest stochastic cost is selected.

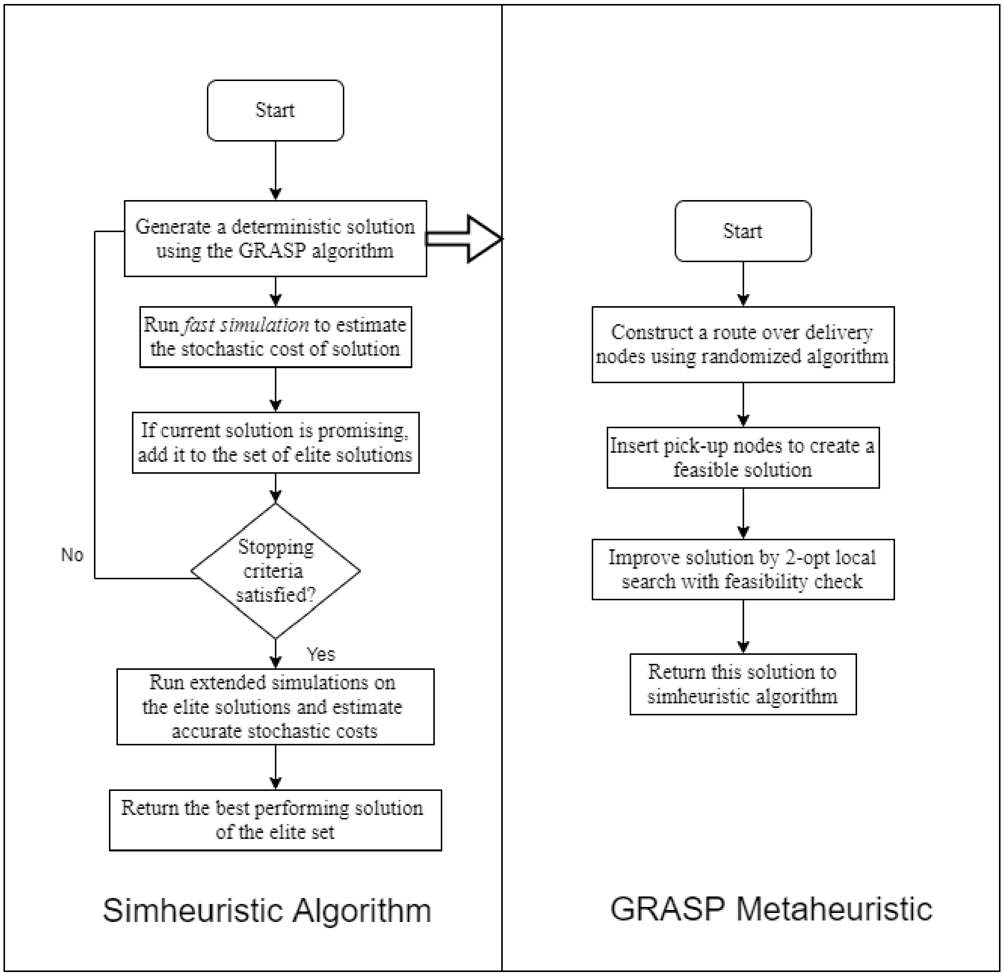

Figure 2: Simheuristic approach.

To reduce the computational time further, we perform the simulation only on the solutions where deterministic cost are within a specific limit of the optimal deterministic cost. For solution x generated by the GRASP heuristic, if the deterministic cost of the resulting solution are within βC^G_{det} (where C^G_{det} is the optimal cost of the deterministic solution), stochastic cost are determined in the second stage using *fast simulation*. Thus, a limited number of simulation runs *mininter* are conducted over solution x. In these runs, the supply and demand values are generated using a distribution with known mean and variance. The vehicle is assumed to follow the route specified by x. As the supply and demand at the nodes may diverge from their expected value, the demands at all locations can not be met if there is not sufficient supply. Let a certain penalty r be associated with each unmet demand. If the load of the vehicle when it visits node i is l_i and s_i is the supply or demand at node i, the penalty cost at node i are calculated as $c^r_i = max(0, r \times (-s_i - l_i))$. Thus, we calculate the penalty of not meeting the demands over all the nodes in a given simulation run k as $C^k_{stoch} = \sum_{i \in N} c^r_i$ and this constitutes the stochastic cost of the given solution. By running the simulation multiple times for each x, the average stochastic cost $C^x_{stoch} = \sum_k C^k_{stoch}/miniter$ are reported as the expected stochastic cost of a given solution. We maintain a set of elite solutions E of fixed size, say K^E, that give lowest stochastic cost. If $|E| < K^E$, x is added to E. If the elite set has reached its capacity, i.e., $|E| \geq K^E$, we check if C^x_{stoch} is smaller than any of those in the elite set. If there exists such a solution x', x replaces x' in E.

This simulation is run for *numiter* iterations (with $numiter > K^E$) and different deterministic solutions that are generated using a randomized heuristic. Once a complete set of elite solutions is obtained, an extended number of $maxiter$ simulation runs are executed on each $e \in E$ to estimate the accurate simulation cost for each element of the elite solution. The stochastic cost are recomputed for each e. The solution giving the lowest stochastic solution is returned and selected as the final solution e^*. C^H_{det} are the deterministic cost associated with the tour length of e^*. C^H_{stoch} are the stochastic cost of e^* obtained by extended simulations on E.

4 COMPUTATIONAL EXPERIMENTS

4.1 Design of Experiments

We implemented the proposed algorithm in the Python Programming language and ran the experiments on a machine equipped with an Intel Core i5 processor working at 2.5 GHz and with 8 GB Ram. For our computational experiments, the 1-PDTSP benchmark data set proposed by Hernández-Pérez and Salazar-González (2004) was employed with some modifications. We use the *TS2004t2* data set (available at Hernández-Pérez et al. 2020). The given data sets contain data for networks with a number of nodes varying from 20 to 60 and a vehicle capacity varying from 10 to 45, as well as a separate data set where the vehicle capacity is 1,000. For each node size and vehicle capacity combination, eight different network instances are given. Because this work does not aim to compare the performance of the deterministic solutions, the capacity of vehicles, which adds further constraints on the problem, is not explicitly considered. Given the demand and supply values, a vehicle capacity of 1,000 essentially refers to the uncapacitated problem. Thus, for our experiments, we select the data sets with nodes 20, 40, and 60 and vehicle capacity 1,000. The existing data sets have a certain demand and supply associated with the depot, which is set to zero in our experiments, as we assume no supply or demand at the depot. This leads to generating data sets where the net supply is either less or greater than the net demand depending on the value at the depot in the original file. In order to study the effect of demand-supply imbalance, we select two instances for each value of n, for each of the two cases, i.e., $N_P \geq N_D$ and $N_P < N_D$. The details of these data sets are shown in Table 1.

Table 1: Data set information.

	$N_P \geq N_D$				$N_P < N_D$			
n	Supply nodes	Demand nodes	Total supply	Total demand	Supply nodes	Demand nodes	Total supply	Total demand
20	12	8	44	37	11	9	36	43
40	24	16	93	85	21	19	89	93
60	30	30	153	148	32	28	119	126

These data sets assume deterministic supply and demand values. For the stochastic input data with supply and demand denoted by $\bar{S}_i$, we extend the data sets using a lognormal distribution. The supply and demand values at each node given in these data sets are considered to be the mean of the underlying lognormal distribution, $E[\bar{S}_i] = S_i$, and $Var[\bar{S}_i] = k.E[\bar{S}_i]$ is the variance of the lognormal distribution for all nodes $i \in N$ where k is the design parameter. The lognormal distribution is described through the location (μ_i) and scale (σ_i) parameters as follows:

$$\mu_i = ln(E[\bar{S}_i]) - \frac{1}{2}\ ln\Big(1 + \frac{Var[\bar{S}_i]}{E[\bar{S}_i]^2}\Big)$$

$$\sigma_i = \sqrt{ln\Big(1 + \frac{Var[\bar{S}_i]}{E[\bar{S}_i]^2}\Big)}$$

The performance of the algorithm for different variances under different values of $k \in \{0,1,2\}$ is simulated in our experiments.

4.2 Experimental Results

In this section, we compare the solutions for the deterministic GRASP Algorithm (G) described in Section 3.1 and the simheuristic (H) proposed in Section 3.2.

Deterministic GRASP Solution Computation:

- The multi-start GRASP algorithm G is first run for $L = 100$ iterations and the best solution is reported as the deterministic solution.
- The cost corresponding to the vehicle route in the final solution constitute the deterministic cost C^G_{det} of the solution.
- We further run $maxiter = 1,000$ iterations on this solution with the supply and demand values given by the lognormal distribution, and calculate the penalty cost of each simulation run. The average penalty cost are reported as the stochastic cost C^G_{stoch} obtained by the GRASP solution.

Simheuristic Computation:

- $L = 100$ solutions are generated by using the GRASP algorithm.
- If the deterministic cost of a given solution are less than βC^G_{det} where $\beta = 1.2$, then $fast\ simulation$ is executed with $miniter = 300$ iterations. The average penalty cost of all simulation runs are calculated to estimate the stochastic cost of a solution.
- The elite set E of size $K = 10$ with solutions giving minimum stochastic cost is determined.
- Further, extended simulation runs with $maxiter = 1000$ iterations are performed on these solutions, and the stochastic cost are recalculated for the elements of the elite set.
- The route with the minimum stochastic cost is selected and the cost associated with this distance traversed by the tour are reported as C^H_{det} and the stochastic cost estimated by extended simulation are represented as C^H_{stoch}.

The results are presented in Tables 2 and 3. Table 2 presents the solution for cases where $N_P \geq N_D$, and Table 3 does the same for cases where $N_P < N_D$. In both tables, the first column presents the number of nodes in the network. The second and third columns represent C^G_{det} and C^G_{stoch} corresponding to the deterministic GRASP algorithm G. The third and fourth columns present corresponding cost for the simheuristic solution H. The fifth column tabulates the increase in deterministic cost in H as compared to G and the sixth column presents the corresponding decrease in stochastic cost. Finally, the last two columns provide the computation time for G and H for 100 iterations. These tables are horizontally divided in three parts, where each part corresponds to a different value of k.

We observe that the deterministic cost, C^H_{det}, take equal or higher values as compared to C^G_{det} for all network sizes when $k = \{1,2\}$. This difference is seen because the GRASP algorithm minimizes the deterministic cost but the simheuristic tries to minimize the stochastic cost with slightly higher values of deterministic cost. The deterministic cost for both G and H increase with n with increasing route length, but are independent of k as it is calculated using the average demand values without considering the variance, and depend on independent simulation runs in each case.

When $k = 0$, the values for C^G_{stoch} and C^H_{stoch} are 0 for all instances when $N_P \geq N_D$. However, when $N_P < N_D$, C_{stoch} takes positive values even when the variance is zero, because stochastic cost represent the cost involved in unsatisfied demands, and all demands are not met in this case. The stochastic penalty cost increase with increasing n and k. The advantage of using the simheuristic algorithm is clear from the percental gap values in Tables 2 and 3. The percental decrease in stochastic cost (column 6) is higher or comparable to the percental increase in the deterministic cost (column 5) when k= $\{1,2\}$. No trend in increasing gap between the C_{det} or C_{stoch} is observed with either n or k values. Thus, the vehicles travel longer distances under the routes given by the simheuristic algorithm, but the number of demands that are met increases significantly, thus reducing the penalty cost of not meeting stochastic demands with available stochastic supply.

Table 2: Results when net demand < net supply.

$N_P \geq N_D$	GRASP Solution (G)		Simheuristic Solution (H)		Percent Gap		Computation Time	
n	C^G_{det}	C^G_{stoch}	C^H_{det}	C^H_{stoch}	C_{det}	C_{stoch}	G	H
	k=0							
20	4117.29	0.00	4187.57	0.00	1.71	-	18.4	15.2
40	5398.3	0.00	5863.5	0.00	8.62	-	36.2	33.5
60	7580.8	0.00	8495.8	0.00	12.07	-	43.6	46.8
	k=1							
20	4117.29	897.5	4243.2	519.1	3.06	42.16	17.6	17.2
40	5215.52	1029.3	5442.6	1002.2	4.41	2.63	30.0	31.6
60	7743.8	2056	7807.5	1757.6	0.82	14.51	44.4	48.1
	k=2							
20	4117.29	1610.3	4243.2	906.6	3.06	43.7	16.1	15.6
40	5221	2059.7	5402.9	1813	3.49	11.98	39.8	36.8
60	7743.4	3967.5	8211.0	3550.8	6.04	10.50	52.7	45.7

Table 3: Results when net demand > net supply.

$N_P < N_D$	GRASP Solution (G)		Simheuristic Solution (H)		Percent Gap		Computation Time	
n	C^G_{det}	C^G_{stoch}	C^H_{det}	C^H_{stoch}	C_{det}	C_{stoch}	G	H
k=0								
20	4096.2	0	4096.2	0	0.00	-	18.5	17.4
40	5795.5	200	5928.4	200	2.29	-	35.2	31.3
60	6844.4	0	7481.2	0	9.32	-	46.8	49.8
k=1								
20	4906.02	689.4	4906.0	673	0.0	2.38	16.5	16.1
40	5759.08	1490	6314.6	1191	9.65	20.07	34.9	31.4
60	7041.5	1537.1	7628.7	1361.4	8.34	13.46	49.4	51.6
k=2								
20	4906.02	1153.1	4906.0	1055.3	0.00	8.48	19.8	19.6
40	5683.06	2494.7	6134.7	2063.0	7.95	17.30	33.5	34.8
60	7037.4	2820.0	7879.3	2507.9	11.96	11.07	44.9	46.5

In order to understand the impact of a demand-supply imbalance, selective results from Tables 2 and 3 are represented in Figures 3 and 4. Figure 3 shows the variation of deterministic cost obtained by the simheuristic algorithm under the conditions $N_P \geq N_D$ or $N_P < N_D$ when $k = \{0,2\}$. We observe that when $n = 20$, for both $N_P < N_D$ and $N_P \geq N_D$, the values of C^H_{det} are the same for all values of k, indicating little effect of variance on the simheuristic solution for a small-sized network. However, the deterministic cost are higher when $N_P < N_D$. The same is observed when $n = 40$, along with variation with k. However, when $n = 60$, C^H_{det} is higher when $N_P \geq N_D$. A similar trend in variation of deterministic cost with parameter k and the supply-demand imbalance is also seen in the solutions obtained by the GRASP algorithm. The higher deterministic cost can be attributed to the fact that the number of supply nodes is higher than that ofdemand nodes in the given examples. When there is excess supply, the vehicle visits all the demand nodes and enough supply nodes to satisfy these demands. Whereas, when supply is less than demand, the vehicle has to visit all the supply nodes, which are higher in number, thus increasing the total distance to be travelled by the vehicle and associated cost. Thus, the vehicle covers longer paths as compared to when it can skip some supply locations leading to smaller deterministic cost.

Figure 4 plots C^H_{stoch} obtained by solving the simheuristic algorithm for the same parameters. These cost are dependent on stochastic supply and demand values. C^H_{stoch} values are comparable for both cases $N_P \geq N_D$ and $N_P < N_D$ for all values of k, and slightly higher for the case where $N_P < N_D$, except when $n = 60$ and $k = 2$. We also observe an increase in stochastic cost with increasing variance for both cases. As the mean demand is higher than the mean supply when $N_P < N_D$, the stochastic demand and, thus, cost associated with unmet demands are also higher.

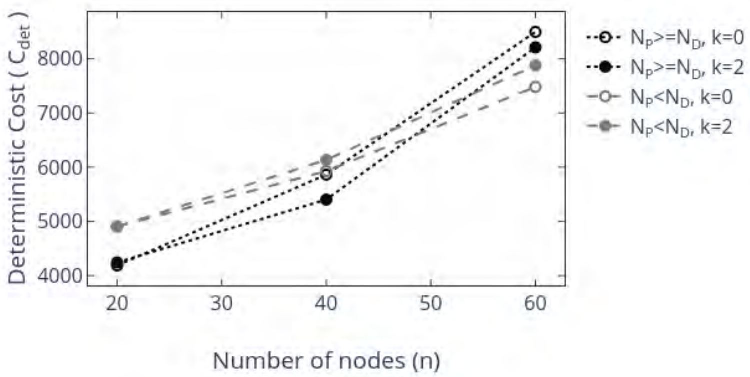

Figure 3: Variation of deterministic cost with n and k (simheuristic approach).

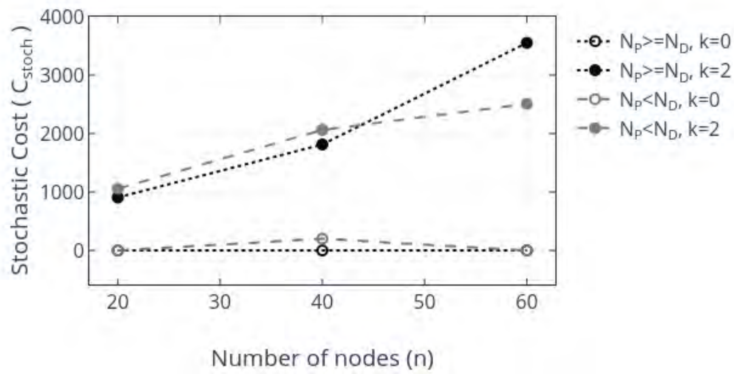

Figure 4: Variation of stochastic cost with n and k (simheuristic approach).

5 CONCLUSION

Motivated by the food redistribution problem, this paper considered the one-commodity selective pick-up and delivery problem (SPDP), which is a novel variant of the one-commodity pick-up and delivery traveling salesman problem. The existing literature on SPDP assumes that all supply and demand quantities are deterministic. However, this may not be the case in practice; e.g., for the food redistribution problem, there may be situations where both demand and supply are stochastic. This paper studied the stochastic SPDP and proposed a simheuristic algorithm to solve it. The resulting algorithm integrates a Monte Carlo simulation into a GRASP metaheuristic framework. The SPDP literature assumes that total supply is always greater than the total demand. We relax this assumption and also study the case where the total supply may not be sufficient to meet the total demand. This study shows that when the demands or supply values are uncertain and the penalty cost are high or the service level is important, simheuristic gives a more dependable solution than the good heuristic solution that does not consider stochastic cost.

Future work is planned. Our work assumes that the travel times between nodes are known with certainty. A natural extension of this work would be to also consider the case where travel times are stochastic. Furthermore, the cases where multiple commodities are transported among pick-up and delivery nodes merit further study.

REFERENCES

Berbeglia, G., J.-F. Cordeau, I. Gribkovskaia, and G. Laporte. 2007. "Static Pickup and Delivery Problems: A Classification Scheme and Survey". *Top* 15(1):1–31.

Festa, P., T. Pastore, D. Ferone, A. A. Juan, and C. Bayliss. 2018. "Integrating Biased-Randomized GRASP with Monte Carlo Simulation for Solving the Vehicle Routing Problem with Stochastic Demands". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. Juan, Mustafee, A. Skoogh, S. Jain, and B. Johansson, 2989–3000. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Gonzalez-Martin, S., B. Barrios, A. Juan, and D. Riera. 2014. "On the Use of Biased Randomization and Simheuristics to Solve Vehicle and Arc Routing Problems". In *Proceedings of the 2014 Winter Simulation Conference*, edited by A.Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 1875–1884. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Guimarans, D., O. Dominguez, J. Panadero, and A. Juan. 2018. "A Simheuristic Approach for the Two-Dimensional Vehicle Routing Problem with Stochastic Travel Times". *Simulation Modelling Practice and Theory* 89:1–14.

Gunes, C., W.-J. van Hoeve, and S. Tayur. 2010. "Vehicle Routing for Food Rescue Programs: A Comparison of Different Approaches". In *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, 176–180. Berlin, Heidelberg: Springer.

Hernández-Pérez, H., and J.-J. Salazar-González. 2004. "Heuristics for the One-Commodity Pickup-and-Delivery Traveling Salesman Problem". *Transportation Science* 38(2):245–255.

Hernández-Pérez, H., I. R. Martín, and J.-J. Salazar-González. 2020. "Pickup-and-Delivery Site". http://hhperez.webs.ull.es/PDsite/#Instances, accessed 31st July 2020.

Ho, S. C., and W. Y. Szeto. 2016. "GRASP with Path Relinking for the Selective Pickup and Delivery Problem". *Expert Systems with Applications* 51:14–25.

Juan, A. A., W. D. Kelton, C. S. Currie, and J. Faulin. 2018. "Simheuristics Applications: Dealing with Uncertainty in Logistics, Transportation, and Other Supply Chain Areas". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. Juan, Mustafee, A. Skoogh, S. Jain, and B. Johansson, 3048–3059. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Maccarrone, L., T. Giovannelli, D. Ferone, J. Panadero, and A. A. Juan. 2018. "A Simheuristic Algorithm for Solving an Integrated Resource Allocation and Scheduling Problem". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. Juan, Mustafee, A. Skoogh, S. Jain, and B. Johansson, 3340–3351. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Pérez, J., Á. Alejandro, F. J. Faulin Fajardo, S. E. Grasman, M. Rabe, and G. Figueira. 2015. "A Review of Simheuristics: Extending Metaheuristics to Deal with Stochastic Combinatorial Optimization Problems". *Operations Research Perspectives* 2:62–72.

Reyes-Rubiano, L., D. Ferone, A. Juan, and J. Faulin. 2019, 06. "A Simheuristic for Routing Electric Vehicles with Limited Driving Ranges and Stochastic Travel Times". *SORT (Statistics and Operations Research Transactions)* 43:1–22.

Ting, C.-K., and X.-L. Liao. 2013. "The Selective Pickup and Delivery Problem: Formulation and a Memetic Algorithm". *International Journal of Production Economics* 141(1):199–211.

AUTHOR BIOGRAPHIES

TEJAS GHORPADE is an M.Tech-Ph.D. dual degree student in the Industrial Engineering and Operations Research Department at IIT Bombay. The main focus of her research includes modelling and simulation of transportation systems with uncertain inputs under dynamic information processes. She has been working on models for different realistic applications like freight rail movements, electric vehicle routing, and container movements. Her email address is trghorpade@iitb.ac.in

CANAN G. CORLU is an Associate Professor of Administrative Sciences Department of the Metropolitan College at Boston University. She holds a Ph.D. in Operations Management from the Tepper School of Business at Carnegie Mellon University. Her research interest is in the area of the design and analysis of stochastic simulation with applications in operations management. Her email address is canan@bu.edu.