Models and Asymptotically Optimal Algorithms for Pickup and Delivery Problems on Roadmaps

Kyle Treleaven, Marco Pavone, Emilio Frazzoli

Abstract— **One of the most common combinatorial problems in logistics and transportation applications is the Stacker Crane problem (SCP), where commodities or customers are associated with a pickup location and a delivery location, and the objective is to find a minimum-length tour visiting all locations with the constraint that each pickup location and its associated delivery location are visited in consecutive order. While vastly many SCPs encountered in practice are embedded in road or road-like networks, very few studies explicitly consider such specific structure. In this paper, first we formulate an environment model that captures the essential features of a "small-neighborhood" road network, including a basic set of road rules. Then, we formulate a *stochastic* version of the SCP on such road network model, where pickup/delivery pairs are random points on network's edges and nodes. Finally, we provide an algorithm for such problem which: (i) is asymptotically optimal, i.e., it produces a solution approaching the optimal one as the number of pickups/deliveries goes to infinity, almost surely; and (ii) is efficient in the sense that it can be computed in time polynomial in n, where n is the number of pickup/delivery pairs. Simulation results show that with a number of pickup/delivery pairs as low as 50 the proposed algorithm delivers a solution whose cost is consistently within 10% of that of an optimal solution.**

I. INTRODUCTION

Pickup and delivery problems (PDPs) constitute an important class of vehicle routing problems in which objects or people have to be transported between locations in a physical environment. These problems arise in many contexts such as logistics, transportation systems, and robotics, among others. Broadly speaking, PDPs can be divided into three classes [1]: 1) Many-to-many PDPs, characterized by several origins and destinations for each commodity/customer; 2) one-to-many-to-one PDPs, where commodities are initially available at a depot and are destined to customers' sites, and commodities available at customers' sites are destined to the depot (this is the typical case for the collection of empty cans and bottles); and 3) one-to-one PDPs, where each commodity/customer has a given origin and a given destination.

When one adds capacity constraints to transportation vehicles, the one-to-one PDP is commonly referred to as the Stacker Crane Problem (SCP). The SCP is a route optimization problem at the core of several transportation systems, including demand-responsive transport (DRT) systems, where users formulate requests for transportation from a pickup point to a delivery point [2], [3].

Literature overview. The SCP, being a generalization of the Traveling Salesman Problem, is NP-Hard [4]. The problem is NP-Hard even on trees, since the Steiner Tree Problem can be reduced to it [5]. In [5], the authors present several approximation algorithms for tree graphs with a worst-case performance ratio ranging from 1.5 to around 1.21. The 1.5 worst-case algorithm, based on a Steiner tree approximation, runs in linear time. Recently, one of the polynomial-time algorithms presented in [5] has been shown to provide an optimal solution on almost all inputs (with a 4/3-approximation in the worst case) [6]. Even though the problem is NP-hard on general trees, the problem is in P on paths [7]. For general graphs, the best approximation ratio is 9/5 and is achieved by an algorithm in [8]. Finally, an average case analysis of the SCP on trees has been examined for the special case of caterpillars as underlying graphs [9].

Despite the importance of the SCP, current algorithms for its solution are either of exponential complexity or come with poor guarantees on their performance in the *worst case*. Yet if the kinds of SCPs that arise in practice are actually comparatively easy (as seems the case for, e.g. the Euclidean TSP), then it should be valuable to find efficient algorithms for such cases with better performance guarantees, e.g. optimality. With this view in mind, the authors have examined the *stochastic* SCP in Euclidean environments [10], where pickup and delivery points are generated according to a i.i.d. random process. In that work the authors were able to produce a polynomial-time algorithm for the SCP that is asymptotically 1-optimal almost surely.

While the former work demonstrates the plausibility of searching for better guarantees for practical cases, it fails to address the most prevalent usage cases of SCPs in modern robotics and transportation contexts, which are usually set in engineered environments characterized by systems of traffic lanes and interchanges. While one might argue that the Euclidean plane can represent such environments at a large scale (e.g. city-wide), if one considers the SCP in smaller, more densely traveled neighborhoods (e.g., the view of a downtown area by taxi or courier service), such a model may fail to capture the essential features of the transportation system.

Contributions. Broadly speaking, the contribution of this paper is two-fold: First, we formulate an environment model that captures the essential features of a continuous "small-neighborhood" road network, including a basic set of *road*

Kyle Treleaven and Emilio Frazzoli are with the Laboratory for Information and Decision Systems, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139 {ktreleav, frazzoli}@mit.edu.

Marco Pavone is with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305 pavone@stanford.edu.

This research was supported in part by the Future Urban Mobility project of the Singapore-MIT Alliance for Research and Technology (SMART) Center, with funding from Singapore's National Research Foundation.

rules (a vehicle constraints model). Despite the prevalence of such environments in the vast majority of modern transportation contexts, a mathematical formalization of such is exceedingly rare. The inclusion of road rules in our model introduces additional complexity in that the way to travel from one set of system coordinates to another may depend on the way a vehicle arrived to the present coordinates in the first place (e.g. which direction the present road was traveled in); this is a consideration not addressed, e.g., with single-integrator vehicle models in $\mathbb{R}^2$. We embed the SCP within a probability framework where origin/destination pairs are identically and independently distributed random variables within the proposed environment. Our random model is general in the sense that we consider potentially non-uniform distributions of points, including the case that the distribution of pickup sites is distinct from that of delivery sites.

Second, we formulate the SCP over points in the environment as an optimization problem; the graph induced by the origin/destination pairs does not have any specific restrictions. The main contribution of the paper is to devise an efficient strategy for solving the SCP—for an embedding of the problem in our road network model—which has strong *probabilistic optimality guarantees*, i.e. is asymptotically optimal almost surely, and which can be computed provably in polynomial time; we provide a preliminary such algorithm.

Organization: The rest of the paper proceeds as follows. In Section II we present background material for the paper. In Section III we formally state the goals of the paper, to devise an efficient strategy to solve the SCP embedded in a road network model. We present our road network model in Section IV. Then in Section V we provide model-specific formulations of the SCP and a closely related combinatorial optimization problem we call the Multi-Crane Problem. In Section VI we present the main result of the paper, an algorithm to solve the stochastic SCP; we prove almost sure asymptotic optimality, and polynomial-time complexity. We present simulation results in Section VII, and offer concluding remarks and direction for future work in Section VIII.

II. Background Material

In this section we present the background material for the paper: we recall some basic geometric definitions, and we present generalized geometrical forms of the Stacker Crane Problem (SCP) and a closely-related problem we call the Multi-Crane Problem.

A. Geometry

Definition 2.1 (Metric space): A *metric space* is the pair of a set Ω, called a *point set*, and a distance function $\|\cdot,\cdot\| : \Omega \times \Omega \to \mathbb{R}_{\geq 0}$, satisfying for all $x_0, x_1, x_2 \in \Omega$:

1) $\|x_0, x_1\| = 0 \implies x_0 = x_1$ (coincidence axiom),
2) $\|x_0, x_1\| = \|x_1, x_0\|$ (symmetry), and
3) $\|x_0, x_1\| \leq \|x_0, x_2\| + \|x_2, x_1\|$ (triangle inequality).

(A *quasi-metric space* is a space that satisfies all of these axioms, except for possibly symmetry.)

Definition 2.2 (Curves, and Uniform Paths): Given a (quasi-) metric space $(\Omega, \|\cdot,\cdot\|)$, a *curve* is a function $\gamma : I \to \Omega$, from some non-empty interval I of the real line to Ω. Given a *continuous* (quasi-) metric space, a uniform path, or simply *path*, is a Lipschitz continuous curve $\mathcal{P} : [0, l) \to \Omega$, for some $l \in \mathbb{R}^+$, for which:

$$\lim_{\delta \to 0^+} \frac{\|\mathcal{P}(x), \mathcal{P}(x+\delta)\|}{\delta} = 1 \qquad \text{for all } x \in [0, l).$$

The scalar l is referred to as the *length* of the path.

Definition 2.3 (Path Fragment): For any path $\mathcal{P} : [0, l) \to \Omega$, a *fragment* of $\mathcal{P}$ is any path (function) resulting from the restriction of $\mathcal{P}$ to some subinterval $[a, b)$ of its domain $(0 \leq a < b \leq l)$.

Definition 2.4 (Path Constraints): Given a (quasi-) metric space $(\Omega, \|\cdot,\cdot\|)$, a set $\mathbb{P}$ of paths over Ω is a *constraints set* if for any $\mathcal{P} \in \mathbb{P}$, all of its fragments, i.e., $\mathcal{P}_{[a,b)}$ for $0 \leq a < b \leq l$, are also in $\mathbb{P}$. The tuple $(\Omega, \|\cdot,\cdot\|, \mathbb{P})$ is called a *path-constrained metric space*. For *continuous* metric spaces, if $\mathbb{P}$ is the set of all uniform paths, then we simply write $(\Omega, \|\cdot,\cdot\|)$; we say that the space is *unconstrained*. For all $x, y \in \Omega$, let $\|x, y\|_{\mathbb{P}}$ denote the infimum length over paths $\{\mathcal{P} \in \mathbb{P} \ : \ \mathcal{P} \text{ from } x \text{ to } y\}$.

B. The Stacker Crane Problem

Given a path-constrained metric space $(\Omega, \|\cdot,\cdot\|, \mathbb{P})$ (or, "environment"), and sets $\mathcal{X}_n := \{x_i\}_{i=1}^n$ and $\mathcal{Y}_n := \{y_i\}_{i=1}^n$ of points in Ω, the Stacker Crane Problem (SCP) is to find:

1) a sequential ordering S of all points with the property that y_i is ordered immediately after x_i for all $i = 1, \ldots, n$; we might interpret each pair (x_i, y_i) as the pick and delivery site, respectively, of some object i; we might interpret S as the order in which the sites are visited by a unit-capacity vehicle, with the intent to pickup and subsequently deliver all such objects;
2) a cyclic path $\mathcal{P} \in \mathbb{P}$ passing through all points in $Q_n := \mathcal{X}_n \cup \mathcal{Y}_n$ in the order of S;

the overall objective of the SCP is to find a path of minimum length satisfying these constraints. We will refer to any such path as a stacker crane tour, and to the sequence S as the *visit sequence*; we refer to any minimum length stacker crane tour as an *optimal* stacker crane tour.

In general, this form of the Stacker Crane Problem is NP-Hard, since its restrictions to various spaces—e.g. to the nodes of a graph [8] or points in Euclidean $\mathbb{R}^d$, $d \geq 1$—recover the traditional graph and Euclidean versions, respectively, of the SCP, which are known to be hard. Like the well-known traveling salesman problem (TSP), the SCP is usually stated in contexts (e.g. the previous two) where the optimal path through a sequence of points is simply the joining of optimal paths between adjacent points along the sequence. We say that such a space is "simple". Our formulation of the SCP does not make the restrictive assumption of simple environments. That is, for a sequence of points $S = (x_1, \ldots, x_n)$, and a set of paths $\{\mathcal{P}_i \in \mathbb{P}\}_{i=1}^{|S|-1}$ where each $\mathcal{P}_i$ is a path between (x_i, x_{i+1}), the path $\mathcal{P}_1 \ldots \mathcal{P}_{|S|-1}$ need not be in $\mathbb{P}$. For the TSP, the lack of such assumptions results in the generalized TSP, or *group* TSP, (see [11], [12]).

In this paper, we are largely interested in a *stochastic* version of the SCP, where the problem instance Q_n is a random set $\{(X_i, Y_i)\}_{i=1}^n$, i.e., X_i and Y_i, for $i = 1, \ldots, n$, are randomly distributed in a continuous environment Ω. We consider the case where all pairs are identically, independently distributed according to a distribution with pdf $\varphi : \Omega \times \Omega \to \mathbb{R}_{\geq 0}$. Throughout the paper we assume that density φ is absolutely continuous. We refer to this construction as the stochastic Stacker Crane Problem; henceforth, we will refer to the stochastic SCP simply as SCP, with the understanding that all pickup-delivery pairs are generated randomly according to the aforementioned probability model.

C. The Multi-Crane Problem

The Multi-Crane Problem (MCP) is closely related to the SCP. The instances of the MCP are the same as those of the SCP; however, the objective is to identify:

1) a partition T of the pairs set $\{(x_i, y_i)\}_{i=1}^n$; and,
2) for each set $t \in T$, a stacker crane tour through t;

the overall objective is to find such a set of stacker crane tours having minimum *total* length $\sum_{t \in T} \mathcal{L}(\mathcal{P}_t)$, where $\mathcal{P}_t$ is the stacker crane tour through t for each subset $t \in T$, and $\mathcal{L}(\cdot)$ gives the length of such a tour. Note that the trivial partition, with a single stacker crane tour (e.g. a solution of the SCP), is a feasible solution for the MCP.

The MCP is different from a problem—we might call it m-SCP—having additional constraint $|T| = m$. The m-SCP arises in a pickup and delivery setting when a *specific* number m of vehicles are available to provide service. The MCP then is the version of the problem where the number of vehicles is not fixed *a priori*, but instead is arbitrary.

III. Problem Statement

In this paper we are interested in the static and stochastic SCP, when pickup and delivery points may be anywhere in a road network (e.g. *between* interchanges); moreover, we consider the case that vehicles are restricted to follow certain *road rules*. Thus, we aim to embed the SCP in a metric space which captures the essential features of a continuous road network. Because an explicit mathematical formalization of such a network, i.e. as a continuous metric space, remains elusive, we aim first to describe the model used in this paper.

Second, given the geometric model, we aim to produce network-specific formulations of the SCP and MCP, and to provide simple reductions of our formulations to well-known combinatorial optimization problems (e.g. TSP).

The ultimate goal of the paper is to find a polynomial-time algorithm for the stochastic SCP which is asymptotically optimal in the strongest probabilistic sense (almost surely); that is, to find an algorithm $\mathcal{A}$ for the SCP, such that

$$\lim_{n \to +\infty} L_{\mathcal{A}} / L^*_{\text{SC}} = 1, \qquad \text{almost surely,}$$

where $L_{\mathcal{A}}$ is the length of the SCP tour produced by algorithm $\mathcal{A}$, and L^*_{SC} is the length of the optimal SCP tour.

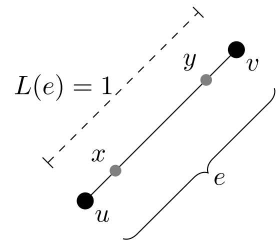


Fig. 1. A simple geometric network; comprises node points u and v, and a single edge e of unit length.

IV. The Geometric Network

A geometric network is a metric space consisting of a continuous, one-dimensional network structure formalizing the notion of a roadmap. The "edges" of the geometric network are continuous segments corresponding to the "roads" of a roadmap. The network "nodes" represent intersections. There is a natural notion of distance between any two locations on a network, which includes, but is not limited to, the notion of distance between *vertices* on their traditional, graphical representations.

The geometric network can be represented by an undirected, non-negatively weighted multigraph $(V, \mathcal{E}, L)$, where V is the vertex set, $\mathcal{E}$ is the edge set, and $L : \mathcal{E} \to \mathbb{R}_{\geq 0}$ is a non-negative length mapping over the set of edges. In the remainder of this section, we will provide the formalism for the geometric network, a metric space, by generating the tuple $(\Omega, \|\cdot, \cdot\|)$ given its representation. The reader should bear in mind that we are only putting into formal terms the most *natural* understanding of a road network geometry. While the development seems to require an unfortunate level of notation, we maintain that the end result should be near to one's initial intuition. Nevertheless, we will make extensive use of illustrations to guide the development.

A. The Point Set

The point set $\mathcal{D}$ of the network is composed of two characteristically distinct sets: the node points, and the edge points. We will refer to all such points in terms of their "addresses" through a symbolic mapping $\mathbf{p}$. Each node $u \in V$ corresponds uniquely to a single point in $\mathcal{D}$, which we denote $\mathbf{p}(u)$; u is the point's address. Together, these are the *node points*.

For example, in Figure 1, the points u and v are node points. Each edge $e \in \mathcal{E}$ corresponds uniquely to some open segment of length $L(e)$ in $\mathcal{D}$; in Figure 1 there is one such segment, having length 1, which connects points u and v. Then for each $0 < t < L(e)$ we denote by $\mathbf{p}(e, t)$ a point which will be understood to lie a distance t away from $\mathbf{p}(u)$ (toward $\mathbf{p}(v)$) along the segment represented by e; (e, t) is its address. Together, these are the edge points. Unfortunately, the preceding informal notion of position for edge points is made ambiguous by the equivalence of the edges (u, v) and (v, u) in the undirected setting. For convenience we assume there is an ordering $\prec$ over the vertex set V, so that in writing $e = (u, v)$ it will be implicit that $u \preceq v$. If $e = (v, u)$ is

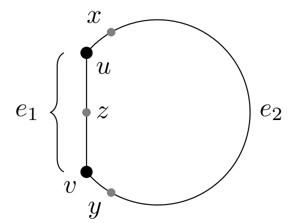


Fig. 2. A ring network; comprises nodes points u and v, with edges e_1 and e_2, of differing length, between them.

written instead, where $v \succ u$, then the appropriate distance reference will be the reverse one. Assuming that $u \prec v$ for the network of Figure 1, then the point x can be written as $\mathbf{p}(e, 0.2)$, and y can be written as $\mathbf{p}(e, 0.8)$. Thus, the network in Figure 1 is represented by a graph with $V = \{u, v\}$ and $\mathcal{E} = \{(u, v)\}$, with a length function L assigning length 1 to the single edge (u, v).

Let $\min_\prec e$ denote the smaller endpoint of e under the given ordering; let $\max_\prec e$ denote the larger endpoint. For all $e \in \mathcal{E}$, let

$$\mathbf{p}(e, 0) := \mathbf{p}(\min\nolimits_\prec e), \text{ and } \mathbf{p}(e, L(e)) := \mathbf{p}(\max\nolimits_\prec e). \quad (1)$$

For example, in Figure 1, the point u can be written as $\mathbf{p}(e, 0)$, and v as $\mathbf{p}(e, 1)$, (since $u \prec v$). Thus, node points generally lack a unique representation; whereas all points described before (1) had been distinct.

We will refer to any set

$$\mathbf{p}(e, (a, b)) := \{\mathbf{p}(e, t) \ : \ t \in (a, b)\}, \quad (2)$$

for $e \in \mathcal{E}$, as an *interval* (open). We refer to the interval $\mathrm{int}(e) := \mathbf{p}(e, (0, L(e)))$ for $e \in \mathcal{E}$ as the *interior* of edge e. The closure $\mathrm{cl}(e)$ of edge e is defined as the interior of e with its endpoints $\{\mathbf{p}(\min_\prec e), \mathbf{p}(\max_\prec e)\}$; note that this definition, along with (1), respects closure of the interval $(0, L(e))$ in the definition of interior using the homeomorphism (2). We define the boundary $\mathrm{bd}(e)$ of edge e as the set of endpoints $\{\mathbf{p}(\min_\prec e), \mathbf{p}(\max_\prec e)\}$; this definition respects the general notion $\mathrm{bd}(\cdot) = \mathrm{cl}(\cdot) - \mathrm{int}(\cdot)$.

B. The Distance Metric

The distance metric for $\mathcal{D}$ can be generated by combination of a set of basic metrics. For each edge $e \in \mathcal{E}$, let

$$\|x, y\|_e := \min\{|t - s| \ : \ x = \mathbf{p}(e, s) \text{ and } y = \mathbf{p}(e, t)\}\,; \quad (3)$$

the $\min$ in (3) ensures a unique distance in the case of points with multiple representations, i.e. node points. We maintain the usual convention $\min \emptyset = +\infty$; thus the metric $\|\cdot, \cdot\|_e$ is finite for $x, y \in \mathrm{cl}(e)$, and infinite otherwise. Collectively, we call these the *edge metrics*. As an example, in Figure 2, $\|x, y\|_{e_2}$ is equal to the distance from x to y in the clockwise direction along e_2. Meanwhile, $\|x, z\|_{e_2}$ is infinite, because the points do not share e_2 as a common edge; note however, that the distance $\|x, u\|_{e_2}$ is finite, as well as the distance $\|u, z\|_{e_1}$. The idea is that for $e \in \mathcal{E}$, and $x, y \in \mathrm{int}(e)$, $\|\cdot, \cdot\|_e$ gives the distance metric on the restriction of $\mathcal{D}$ to $\mathrm{int}(e)$.

We define the *node equality* metric as

$$\|x, y\|_{\mathbf{1}} := \begin{cases} 0 & \text{if } x = y = \mathbf{p}(u),\ u \in V, \\ +\infty & \text{otherwise.} \end{cases}$$

Then the distance metric on $\mathcal{D}$ is equal to

$$\|x, y\|_{\mathcal{D}} := \min_S \sum_{i=1}^{n-1} \min\left\{\|s_i, s_{i+1}\|_{\mathbf{1}}, \min_{e \in \mathcal{E}} \|s_i, s_{i+1}\|_e\right\}, \quad (4)$$

where $S = (s_1, \ldots, s_n)$ may be any finite sequence of points such that $s_1 = x$ and $s_n = y$; if equation (4) is finite, we say the points are *connected*. If two points are connected in the network, then a minimizing sequence for (4) can always be found where the subsequence $(s_2, \ldots, s_{n-1})$ (which may be empty) is composed of only node points. Returning to the example of Figure 2, one could show that $\|x, z\|_{\mathcal{D}} = \|x, u\|_{e_2} + \|u, z\|_{e_1}$, and thus is finite; one could also argue that $\|x, y\|_{\mathcal{D}} = \|x, u\|_{e_2} + \|u, v\|_{e_1} + \|v, y\|_{e_2}$ (the *counter*-clockwise path), and observe that $\|x, y\|_{\mathcal{D}} < \|x, y\|_{e_2}$.

C. A Probability Model for Random Points

This paper is about the properties of a number of combinatorial optimization problems whose instances are randomly-generated sets of points on a geometric network. We use a probability model with sample space $\mathcal{D}$ (the network), and a set of measurable events given by the Borel σ-algebra $\mathfrak{F}$ over node points and closed intervals on edges. For point random variables with support only over edges, we describe probability distributions by their probability density functions (pdf); the notion of probability density extends in a straightforward way from the notion of density over regular line segments, i.e. $\varphi_X(\mathbf{p}(e, y)) := \lim_{\Delta \to 0} \mathbb{P}[X \in \mathbf{p}(e, [y, y + \Delta])]/\Delta$.

D. The Parisian Constraints Model

In this section we introduce a basic model of *road rules* on geometric networks, which we call *Parisian path constraints*. Such constraints are meant to represent abstractly the kinds of driving restrictions that exist in real road networks; the name is given in recognition of the model's capturing the essence of rotary interchanges ("round-abouts"), as prevalent in the road systems of French cities and towns.

Definition 4.1 (Parisian path constraints): Given a geometric network $(\mathcal{D}, \|\cdot\|_{\mathcal{D}})$, the set $\mathbb{F}$ of *Parisian paths* is defined recursively as follows.

1) All direct paths from endpoint to endpoint along a single edge are in $\mathbb{F}$; i.e., $\mathcal{P}(t)$ is defined over $[0, L(e))$ for some edge e; and $\mathcal{P}(t) = \mathbf{p}(e, t)$, or else $\mathcal{P}(t) = \mathbf{p}(e, L(e) - t)$.
2) All uniform paths formed by concatenation of paths of the former type are in $\mathbb{F}$;
3) To satisfy the main property of a constraints set, all fragments of paths of the former types are in $\mathbb{F}$.

The set of Parisian paths is distinct from the set of all uniform paths only by the ordering of rules (2) and (3) of its construction. The appropriate intuition about Parisian path constraints is that once a vehicle begins to travel along an edge in a given direction, it is constrained to complete the traversal of that edge in the same direction, until it arrives at the opposite endpoint; whereas the direction of travel from node points is arbitrary. A Parisian path-constrained environment $(\mathcal{D}, \mathbb{F})$ is *not* simple. For example, consider two points x and y in close proximity along some network edge. The shortest (Parisian) path from x to y is the direct one, as is the shortest path from y to x, yet the shortest path through the sequence (x, y, x) is different from their joining.

Definition 4.2 (State-space): Given a path constrained environment $(\Omega, \mathbb{P})$, a state-space is a *quasi-metric* space $(X, \|\cdot\|_X)$ with an *projection* function $\mathrm{proj}_\Omega : X \to \Omega$, having the property that the image of the set of uniform paths over X, under projection proj_Ω, is equal to the constraints set $\mathbb{P}$ over Ω.

Given an environment $(\Omega, \mathbb{P})$, the motivation behind constructing a state-space X is to recover the desirable properties of a *simple* environment. State-spaces by design enjoy a Markov-like property: Given a path prefix $\mathcal{Q}'$ terminating in state q, the set of allowable extensions of $\mathcal{Q}'$ is simply all uniform extensions, i.e., it is independent of the prefix $\mathcal{Q}'$.

We introduce the following realization of a Parisian state-space (we call it $\mathcal{F}$), given a geometric network $\mathcal{D}$. Our construction is based on sufficient statistics for satisfying the Markov property: Specifically, to extend any Parisian path it is sufficient to know the path's endpoint and, if it is an edge point, the current direction of travel. Thus, for each $u \in V$, let $\mathbf{q}(u)$ denote the (unique) state at point $\mathbf{p}(u)$ (no direction). For each $e \in \mathcal{E}$, and $0 < t < L(e)$, let $\mathbf{q}^+(e,t)$ denote the state at point $\mathbf{p}(e,t)$ with travel in the *increasing* direction (according to $\prec$); let $\mathbf{q}^-(e,t)$ denote the state at the same point with travel in the *decreasing* direction. We institute the familiar convention $\mathbf{q}^\pm(e,0) := \mathbf{q}(\min_\prec e)$, and $\mathbf{q}^\pm(e, L(e)) := \mathbf{q}(\max_\prec e)$, for all $e \in \mathcal{E}$, i.e. node states do not have a unique address. The state space is completed by:

1) distance quasi-metric $\|\cdot,\cdot\|_\mathcal{F}$ in the form of (4), derived from *edge quasi-metrics*
$$\|g,h\|_e := \min\left\{ |t-s| \;:\; \substack{g=\mathbf{q}^+(e,s)\\ h=\mathbf{q}^+(e,t),\\ s<t} \text{ or } \substack{g=\mathbf{q}^-(e,s)\\ h=\mathbf{q}^-(e,t)\\ s>t} \right\}$$
for all $e \in \mathcal{E}$, and the same node equality metric;
2) projection function $\mathrm{proj}_\mathcal{D}$, such that $\mathrm{proj}_\mathcal{D}(\mathbf{q}(u)) = \mathbf{p}(u)$ for all $u \in V$, and $\mathrm{proj}_\mathcal{D}(\mathbf{q}^\pm(e,t)) = \mathbf{p}(e,t)$ for all $e \in \mathcal{E}$, and $0 \le t \le L(e)$.

V. The SCP Over Geometric Networks with Parisian Path Constraints

In simple environments, the SCP has a straightforward reduction to the TSP. Given a SCP instance Q_n, the ATSP problem graph $G = (V, A)$ contains one node for each *demand*, and edge weights (i.e. distances) given by $d_{u^i u^j} := \|y_i, x_j\|_\mathbb{P} + \|x_j, y_j\|_\mathbb{P}$, for nodes $u^i, u^j \in V$ corresponding to demands i and j, respectively; $d_{u^i u^j}$ is the length of a minimum-length path through the sequence of points (y_i, x_j, y_j) (a *delivery-to-delivery* tour fragment); The optimal TSP tour of G gives the visit sequence of the optimal SCP through Q_n; the SCP tour itself can be stitched together from the optimal paths between points along this sequence. Unfortunately, such stitching may not produce a valid path for *non-simple* geometries.

SCP Formulation over Geometric Networks with Parisian Path Constraints: For the (non-simple) geometric network with Parisian path constraints, we provide a reduction of the SCP to the generalized traveling salesman problem [11], or *group* TSP. The group TSP is, given a possibly asymmetric distance graph, and a partition of the graph nodes into disjoint subsets $\{S_i\}$, or *groups*, to find the shortest cyclic tour visiting exactly one element from each group. The group TSP can be formulated in the style of [11]:

Let x_{ij}, for all $(i,j) \in A$, be the binary variables indicating the presense (if $=1$), or absense (if $=0$), of edge (i,j) in the group TSP tour. The group TSP polytope can be described by:

$$\sum_{i\,:\,(i,k)\in A} x_{ik} = \sum_{j\,:\,(k,j)\in A} x_{kj} \qquad (k \in V) \tag{5}$$

$$\sum_{(i,k)\in A\,:\,k\in S_l} x_{ik} = \sum_{(k,j)\in A\,:\,k\in S_l} x_{kj} = 1 \quad (l = 1,\dots,n) \tag{6}$$

$$\sum_{(i,j)\in A\,:\,i,j\in T} x_{ij} \le |T| - 1 \tag{7}$$
$$T \subseteq V, T \cap S_l \neq 0 \text{ for some but not all } l.$$

The group TSP is to minimize $\sum_{(i,j)\in A} d_{ij} x_{ij}$, subject to (5), (6), and (7), where $\{d_{ij} : i,j \in V\}$ are edge costs.

Given a SCP instance Q_n, the problem graph contains one *group* S_i for each demand i. For each demand i, S_i contains one node for each element of $\{\mathbf{q}^\pm(x_i)\} \times \{\mathbf{q}^\pm(y_i)\}$; that is, all pairs (g,h) such that a (state-space) path exists from x_i in state g, to y_i in state h. The edge weights are given by

$$\{d_{uv} := \|h_u, g_v\|_\mathcal{F} + \|g_v, h_v\|_\mathcal{F} \;:\; u, v \in V\}, \tag{8}$$

where u and v are nodes in V corresponding to pairs (g_u, h_u) and (g_v, h_v), respectively; such weights are, by the Markov property, the lengths of the optimal tour fragments through sequences of the form (h_u, g_v, h_v).

The idea behind this reduction is that if a group TSP tour visits group S_i by the node corresponding to state pair (g,h), then the corresponding stacker crane tour will service the demand i by performing the pickup at x_i in state g, followed by delivery at y_i in state h (generally using the optimal path to do so). The visit sequence of the optimal group TSP (an ordering of groups), gives the visit sequence of the optimal SCP (an ordering of demands). To show correctness, we observe the following fact.

Lemma 5.1: If $\mathcal{P}$ is a (Parisian) stacker crane tour on a geometric network, then there exists a TSP tour $\hat{\sigma}$ on the related instance of the group TSP that has cost no greater than the length of $\mathcal{P}$. Moreover, any group TSP tour can be transformed into a stacker crane tour on the network of length equal to its cost.

Proof: Let σ be the visit sequence of cyclic Parisian tour $\mathcal{P}$, with a corresponding state-space curve $\mathcal{Q}$. To construct a TSP tour $\hat{\sigma}$ for the group TSP instance, we traverse the cycle $\mathcal{Q}$ once, starting from some delivery point. (Let $\hat{\sigma}$ be initially an empty graph.) For each pair of demands, say (i, j), that are adjacent in σ—including the pair of the cycle boundary—we will add an edge (u, v) to $\hat{\sigma}$: The node u is the graph node in S_i associated with a state pair (g_i, h_i), where g_i is the state by which $\mathcal{Q}$ visits x_i, and h_i is the state by which $\mathcal{Q}$ visits y_i; the node v is chosen in the same way for demand j. Note that the total weight of edges added to $\hat{\sigma}$ in this way is bounded above by $\mathcal{L}(\mathcal{P})$: Between the delivery of demand i and that of demand j, $\mathcal{Q}$ traverses a path through a sequence of states (h_i, g_j, h_j). The edge weights (8) of G are chosen as the lengths of *optimal* such paths. It remains only to show that the subgraph $\hat{\sigma}$ is a feasible group TSP tour. Each *demand* is visited exactly once by $\mathcal{Q}$, therefore each *group* is visited exactly once by $\hat{\sigma}$.

The second part of the proof is, roughly speaking, by reversing the previous procedure. Let $\hat{\sigma}$ be a group TSP tour on G. We traverse the tour once. For each arc (u, v), we add to $\mathcal{P}$ an optimal delivery-to-delivery tour fragment through the sequence of states (h_i, g_j, h_j), where demands i and j (and states g_i, h_i, g_j and h_j) correspond to u and v in the previous sense. The result is a Parisian stacker crane tour of length equal to the cost of $\hat{\sigma}$. ■

MCP Formulation over Geometric Networks with Parisian Path Constraints: Recall the previous group TSP formulation of the SCP; let us call it Problem A. Now, for the geometric network with Parisian path constraints, we provide a reduction of the MCP to another graphical optimization problem. Let us consider Problem B, the relaxation of Problem A produced by neglecting the integrality constraints (i.e., all $x_{ij} \in \{0, 1\}$), as well as the constraints of equation (7), or the so-called *subtour elimination constraints* (see [11]). All of the *integer* solutions of Problem B correspond to subgraphs of G, where (i) the in-degree and out-degree of every node are the same (equation (5)), and (ii) there is exactly one edge entering (or leaving) each group S_i, for $i = 1, \ldots, n$ (equation (6)). While the authors are not aware of the existence of a specialized algorithm for the MCP, we now show that the problem can be solved in polynomial time by solving Problem B (a linear program).

Lemma 5.2: If $\{\mathcal{P}_t\}_{t \in T}$ is a solution to the MCP, i.e. a set of $|T|$ stacker crane tours visiting all demands, then there is a solution $\hat{\sigma}$ to Problem B having cost no greater than the total length $\sum_{t \in T} \mathcal{L}(\mathcal{P}_t)$. Moreover, given a solution $\hat{\sigma}$ to Problem B, a solution to the MCP can be constructed with total length equal to its cost.

Proof: The proof of the first part is by construction of a feasible subgraph $\hat{\sigma}$ of G, given any MCP solution. (Let $\hat{\sigma}$ be initially an empty graph.) For each stacker crane tour $\mathcal{P}_t$ (for $t \in T$), we apply the procedure in the proof of Lemma 5.1 (though keeping our "progress" in $\hat{\sigma}$). Using the same argument, we have that the total weight of edges added to $\hat{\sigma}$ in this way is bounded above by $\sum_{t \in T} \mathcal{L}(\mathcal{P}_t)$. It remains only to show that $\hat{\sigma}$ is a feasible solution to Problem B. Let $\hat{\sigma}_t$ denote the part of $\hat{\sigma}$ contributed by the tour $\mathcal{P}_t$, for each $t \in T$. Because the tours $\{\mathcal{P}_t\}_{t \in T}$ visit all the demands disjointly, any group S_i contains edges from exactly one $\hat{\sigma}_t$. By the properties of the procedure, any $\hat{\sigma}_t$ visits its assigned groups exactly once, and each by a single node. Thus, each group has a single incoming and a single outgoing edge (satisfying (i)), and each node has equal in-degree and out-degree; specifically, both $= 1$ if the node is visited, or $= 0$ otherwise (satisfying (ii)).

The second part of the proof is again by essentially reversing the procedure from the first part; however, this requires that a solution σ of Problem B have integer components, i.e. all $x_{ij} \in \{0, 1\}$. Luckily, an extension of the Birkhoff-von Neumann theorem [13] ensures that any feasible solution of Problem B can be written as a convex combination of integral solutions; any integral component of an optimal solution of Problem B is itself an optimal solution. Focusing then on such solutions, and combining (5) and (6), we have that exactly one node in each group has positive degree, with one incoming edge and one outgoing edge. The positive degree nodes of such a graph (one for each group) can be partitioned to form a set of disjoint cycles. To construct a feasible MCP solution, we produce one stacker crane tour for each cycle, using the reverse procedure in the proof of Lemma 5.1.

■

VI. An Asymptotically Optimal Polynomial-Time Algorithm for the SCP

In this section we present the main result of the paper, an algorithm for the stochastic SCP that is asymptotically optimal almost surely. We proceed with a formal description of the algorithm. Then we discuss the computational complexity of the algorithm. Finally we prove its almost sure, asymptotic optimality.

The key idea behind the algorithm is to service the demands of the randomly-generated SCP instance by traversing each of the stacker crane tours in the optimal MCP solution. Unfortunately, the tours in the MCP are generally disjoint, and so cannot be traversed by a single vehicle without adding path segments of additional length. On the other hand, it is easy to show that the MCP tours can be stitched into a traversable path (stacker crane tour), where the total length of additional segments is bounded above by a constant; this constant is a function of the shape of the network only, and does not scale with the size of the problem.

Suppose we have a finite-length closed-path $\mathcal{P}_c$ visiting every state in $\mathcal{F}$; $\mathcal{F}$ is one-dimensional and finite length itself, so such a covering should exist. For example, in Figure 3, a network is shown with five nodes arranged in a regular pentagon. The closed Parisian tour drawn around its periphery, i.e. $(1, 2, 3, 4, 5, 1, 5, 4, 3, 2, 1)$, covers every possible vehicle state; it traverses the entire ring once in the clockwise direction, and then again in the counter-clockwise direction. The (optimal) MCP tours can be transformed into a single stacker crane tour using the following rule:

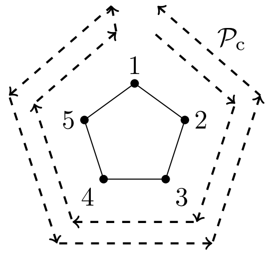


Fig. 3. A ring network with five (5) evenly-spaced interchanges. A state-covering tour $(1, 2, 3, 4, 5, 1, 5, 4, 3, 2, 1)$ is drawn.

The Stitching Algorithm:

1) Traverse the closed-path $\mathcal{P}_{\rm c}$, from any starting point, until some pickup is reached in a state g by which it is visited by one of the optimal MCP tours (i.e. the first such pickup).
2) Let $\mathcal{P}_t$ be the visiting tour. Service all of the demands on the closed tour $\mathcal{P}_t$ (traversing it), returning eventually to state g.
3) Continue on $\mathcal{P}_{\rm c}$ from state g, servicing all of the (remaining) unserviced MCP tours in the same way, returning eventually to the starting state.

While the MCP can be solved in polynomial time, e.g. using a general purpose LP solver for Problem B, it is unlikely that it can be solved in time $o(n^3)$ without significant effort: There is a straightforward $O(n)$-reduction of the bipartite matching problem (BMP) to MCP, and the fastest known BMP algorithm on general graphs [14] is $O(|V||A|)$, i.e. n^3. In other words, an $o(n^3)$ algorithm for MCP would immediately improve the state of the art for BMP. Thus, it is fairly certain that the time complexity of the stitching algorithm is dominated by the construction of the optimal MCP tours. Sorting the n pickup states by order of occurence along $\mathcal{P}_{\rm c}$ can be achieved in time $O(n \log n)$, and constructing the final tour takes $O(n)$.

Theorem 6.1: Let $Q_n = \{(X_i, Y_i)\}_{i=1}^n$ be a stochastic instance of the SCP in $\mathcal{D}$ with Parisian path constraints $\mathbb{F}$, where (X_i, Y_i) are identically, independently distributed according to a distribution with density φ. Let $L^*_{\rm SC}$ be the length of the optimal stacker crane tour; let $L_{\rm stitch}$ be the length of the stacker crane tour generated by the stitching algorithm. Then

$$\lim_{n\to\infty} L_{\rm stitch}/L^*_{\rm SC} = 1, \qquad \text{almost surely.}$$

Proof: Let $\{\mathcal{P}_t\}_{t\in T}$ be an optimal solution to the MCP over Q_n, i.e. a set of stacker crane tours collectively visiting all points. Let $\mathcal{P}_{\rm c}$ be a finite-length state covering of the network, used by the stitching algorithm. The stacker crane tour generated by the stitching algorithm has length $L_{\rm stitch}$, where

$$L^*_{\rm SC} \le L_{\rm stitch} = \sum_{t\in T} \mathcal{L}(\mathcal{P}_t) + \mathcal{L}(\mathcal{P}_{\rm c}) \le L^*_{\rm SC} + \mathcal{L}(\mathcal{P}_{\rm c});$$

we have the equality because the tour $\mathcal{P}_{\rm c}$ is traversed exactly once to connect the MCP subtours; we have the last inequality because the MCP is a relaxation of the SCP. Thus, we have $\lim_{n\to\infty} L_{\rm stitch}/L^*_{\rm SC} \le \lim_{n\to\infty} 1 + \mathcal{L}(\mathcal{P}_{\rm c})/L^*_{\rm SC}$, where $\mathcal{L}(\mathcal{P}_{\rm c})$ is constant. Observing that the optimal stacker crane tour has length that grows linearly in n (almost surely), e.g. because $\mathbb{E}\left[\|X_i, Y_i\|_{\mathcal{D}}\right] > 0$ (and applying Strong Law of Large Numbers), we obtain the limit. ■

An Asymptotically Optimal Multiple Vehicle Makespan Algorithm: It is worth noting that the asymptotically optimal SCP algorithm of this paper can be used to solve another related "Stacker Crane"-*like* problem. Let us call it the m-Stacker Crane Makespan Problem (m-SCMP). The m-SCMP is to find

1) a partition T of the pairs set $\{(x_i, y_i)\}_{i=1}^n$, where $|T| = m$; and,
2) for each set $t \in T$, a stacker crane tour through t;

the objective of the m-SCMP is to minimize the *maximum* length ($\max_{t\in T} \mathcal{L}(\mathcal{P}_t)$) among the set of stacker crane tours produced; note that the objective differs from the *total* length objective of the m-SCP. The approach to constructing an asymptotically optimal m-SCMP algorithm (for fixed m) using an asymptotically optimal algorithm for the 1-SCP, is to break the resulting stacker crane tour into m equal-length fragments (or approximately equal); this approach has been described in [8]. The makespan version of the SCP is particularly important in dynamic multi-vehicle versions of the problem, as discussed, e.g. in [15].

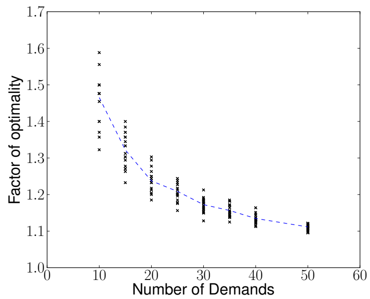
VII. Simulation Results

In this section we present an empirical study, through randomized simulations, of the performance of the stitching algorithm; we consider performance in the sense of quality of the solution, as well as runtime of the algorithm. For our experiments, we used the simple geometric network shown in Figure 3, with Parisian path constraints. All points (pickup and delivery) were sampled i.i.d. according to a uniform distribution over the entire length of the network, i.e. $\varphi(x) = 1/5$ for all $x \in \mathcal{D}$.

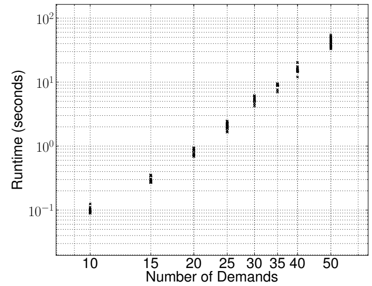
Figure 4(a) shows a plot of the ratios $L_{\rm stitch}/\sum_{t\in T} \mathcal{L}(\mathcal{P}_t)$ observed for a set of random samples. We used twenty (20) samples in each of a number of size categories. The factor plotted is an upper bound on the *factor of optimality* $L_{\rm stitch}/L^*_{\rm SC}$; for instances of the size considered, it is impractical for us to compute optimal stacker crane tours explicitly. The plot shows a trend decreasing as $O(1/n)$ (as expected), and having sub-optimality consistently less than 20% for > 30 demands, and just above 10% for ≈ 50 demands.

Figure 4(b) shows the total runtime of the algorithm, on the same set of samples. The observed growth in runtime indicates that our preliminary MCP algorithm, based on using a general-purpose LP solver (GNU Linear Programming Kit (GLPK) software, glpsol), has runtime of the order $n^{\approx 4}$.

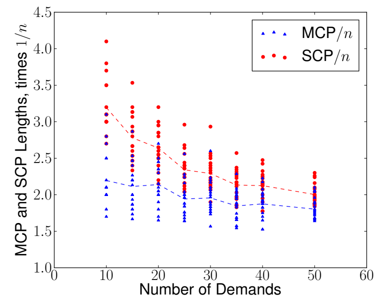
Figure 4(c) displays the factors $\sum_{t\in T} \mathcal{L}(\mathcal{P}_t)/n$ and $L_{\rm stitch}/n$—per-demand average tour lengths—for again the same set of samples. Dashed lines are drawn through the mean ratio of each size category. As n grows, such factors must be bounded below almost surely by a constant $C \ge \mathbb{E}\left[\|X_i, Y_i\|_{\mathcal{D}}\right] = 1.75$; our conjecture, which we leave to



(a) The factor L_{stitch} vs. MCP.



(b) Runtime of the stitching algorithm.



(c) *Per-demand* average MCP and SCP lengths.

discuss in a future paper, is that the factors converge to C, and that the inequality is strict.

VIII. Conclusion

In this paper we have formulated the Stacker Crane Problem in the stochastic setting for an environment model designed to capture the essential features of the ubiquitous road network. Given the model, we have provided a polynomial-time algorithm, based on the Multi-Crane Problem, which is asymptotically optimal for the SCP almost surely. We believe that the MCP-based algorithm is a significant contribution, because it has been non-trivial to produce efficient policies having asymptotic optimality guarantees, even in the stochastic case. There are a number of simple strategies for the SCP that are constant-factor optimal, i.e. $\lim_{n\to\infty} L_{\mathcal{A}}/L^*_{\text{SC}} = C$ for some $C \geq 1$, but whose constants C are, *provably*, strictly greater than one. One class of algorithms foremost among such "naive" algorithms is the class of algorithms that approach the stacker crane problem by first determining optimal paths from the pickup points to their matched delivery points (to simplify the problem) and then constructing a stacker crane tour from these fragments (e.g. by an algorithm [10] based on bipartite matching). It can be shown that the set of such paths are at best constant-factor optimal, with $C > 1$; we leave a detailed discussion for an upcoming paper.

While constant factor guarantees may suffice for small instances in undemanding problem domains, a large body of research is dedicated to *dynamic* vehicle routing problems [16] for which they do not. For example, suppose we have a spatial queuing system, with service of a pickup-delivery type; demands arrive to the system over time, according to a renewal process, say at rate λ, and a finite number of vehicles are routed to provide service. The previous guarantees do not lend themselves to proving conditions (e.g., bounds on the arrival rate λ) that are necessary and sufficient to ensure a stable queueing system, nor to creating efficient stabilizing policies. The guarantees provided by our algorithm have such ability, which we will explore in future papers.

References

[1] G. Berbeglia, J.-F. Cordeau, I. Gribkovskaia, and G. Laporte. Static pickup and delivery problems: a classification scheme and survey. *TOP*, 15:1–31, 2007. 10.1007/s11750-007-0009-0.

[2] J. F. Cordeau, G. Laporte, J. Y. Potvin, and M.W.P. Savelsbergh. Transportation on demand. In C. Barnhart and G. Laporte, editors, *Transportation, Handbooks in Operations Research and Management Science*, volume 14, pages 429–466. Elsevier, Amsterdam, The Netherlands, 2007.

[3] W. J. Mitchell, C. E. Borroni-Bird, and L. D. Burns. *Reinventing the Automobile*. MIT Press, 2010.

[4] G. N. Frederickson and D. J. Guan. Preemptive ensemble motion planning on a tree. *SIAM Journal on Computing*, 21(6):1130–1152, 1992.

[5] G. N. Frederickson and D. J. Guan. Nonpreemptive ensemble motion planning on a tree. *J. Algorithms*, 15:29–60, July 1993.

[6] A. Coja-Oghlan, S. O. Krumke, and T. Nierhoff. A heuristic for the stacker crane problem on trees which is almost surely exact. *Journal of Algorithms*, 61(1):1 – 19, 2006.

[7] M. J. Atallah and S. R. Kosaraju. Efficient solutions to some transportation problems with applications to minimizing robot arm travel. *SIAM Journal on Computing*, 17(5):849–869, 1988.

[8] G. N. Frederickson, M. S. Hecht, and C. E. Kim. Approximation algorithms for some routing problems. *Foundations of Computer Science, Annual IEEE Symposium on*, 0:216–227, 1976.

[9] A. Coja-Oghlan, S. O. Krumke, and T. Nierhoffi. Scheduling a server on a caterpillar network - a probabilistic analysis. In *Proceedings of the 6th Workshop on Models and Algorithms for Planning and Scheduling Problems*, 2003.

[10] K. Treleaven, M. Pavone, and E. Frazzoli. Asymptotically optimal algorithms for pickup and delivery problems with application to Large-Scale transportation systems. *arXiv:1202.1327*, February 2012.

[11] G. Laporte, H. Mercure, and Y. Nobert. Generalized travelling salesman problem through n sets of nodes: the asymmetrical case. *Discrete Applied Mathematics*, 18(2):185–197, 1987.

[12] G. Laporte, A. Asef-Vaziri, and C. Sriskandarajah. Some applications of the generalized travelling salesman problem. *Journal of the Operational Research Society*, pages 1461–1467, 1996.

[13] L. Mirsky. Results and problems in the theory of doubly-stochastic matrices. *Probability Theory and Related Fields*, 1(4):319–334, 1963.

[14] Z. Galil, S. Micali, and H. Gabow. Priority queues with variable priority and an o(ev log v) algorithm for finding a maximal weighted matching in general graphs. pages 255 –261, 1982.

[15] M. R. Swihart and J. D. Papastavrou. A stochastic and dynamic model for the single-vehicle pick-up and delivery problem. *European Journal of Operational Research*, 114(3):447–464, 1999.

[16] P. Toth and D. Vigo. *The vehicle routing problem*, volume 9. Society for Industrial Mathematics, 2002.