Multi-Vehicle Routing

Emilio Frazzoli (MIT)   Marco Pavone (Stanford)

Abstract

Multi-vehicle routing problems in systems and control theory are concerned with the design of control policies to coordinate several vehicles moving in a metric space, in order to complete spatially localized, exogenously generated tasks, in an efficient way. Control policies depend on several factors, including the definition of the tasks, of the task generation process, of the vehicle dynamics and constraints, of the information available to the vehicles, and of the performance objective. Ensuring the stability of the system, i.e., the uniform boundedness of the number of outstanding tasks, is a primary concern. Typical performance objectives include measures of quality of service, such as, e.g., the average or worst-case time a task spends in the system before being completed, or the percentage of tasks that are completed before certain deadlines. The scalability of the control policies to large groups of vehicles often drives the choice of the information structure, requiring distributed computation.

Keywords and Phrases: multi-vehicle routing, dynamic routing, task allocation, networked robots, cooperative control, decentralized control.

Introduction

Multi-vehicle routing problems in systems and control theory are concerned with the design of control policies to coordinate several vehicles moving in a metric space, in order to complete spatially localized, exogenously generated tasks, in an efficient way. Key features of the problem are that tasks arrive *sequentially* over time and planning algorithms should provide *control policies* (in contrast to pre-planned routes) that prescribe how the routes should be updated as a function of those inputs that change in real-time. This problem is usually referred to as Dynamic Vehicle Routing (DVR). In DVR problems, ensuring the stability of the system, i.e., the uniform boundedness of the number of outstanding tasks, is a primary concern.

Motivation and Background

As a motivating example, consider the following scenario: a team of Unmanned Aerial Vehicles (UAVs) is responsible for investigating possible threats over a region of interest. As possible threats are detected, by intelligence, high-altitude or orbiting platforms, or by ground sensor networks, one of the UAVs must visit its location and investigate the cause of the alarm, in order to enable an appropriate response if necessary. Performing this task may require the UAV not only to fly to the possible threat's location, but also to spend additional time on site. The objective is to minimize the average time between the appearance of a possible threat and the time one of the UAVs completes the close-range inspection task. Variations may include priority levels, time windows during which the inspection task must be completed, and sensors with limited range.

In order to perform the required mission, the UAVs (or, more in general, mission control) need to repeatedly solve three *coupled* decision-making problems:

1. **Task allocation:** Which UAV shall pursue each task? What policy is used to assign tasks to UAVs? How often should the assignment be revised?,
2. **Service scheduling:** Given the list of tasks to be pursued, what is the most efficient ordering of these tasks?
3. **Loitering paths:** What should UAVs without pending assignments do?

DVR problems, including the above UAV routing problem, are generally *intractable* due to their multi-faceted combinatorial and stochastic nature, and consequently solution approaches have been devised that look either at heuristic algorithms or at approximation algorithms with some guarantee on their performance.

Related problems

DVR problems represent the dynamic counterpart of the well-known static vehicle routing problem (VRP), whereby: (i) a team of *m* vehicles is required to service a set of *n* "static" tasks in a metric space; (ii) each task requires a certain amount of on-site service; (iii) the goal is to compute a set of routes that minimizes the cost of servicing the tasks; see Toth and Vigo [2001] for a thorough introduction to this problem. The VRP is *static* in the sense that vehicle routes are computed assuming that no new tasks arrive. The VRP is an important research topic in the operations research community.

Approaches for Multi-Vehicle Routing

Broadly speaking, there are three main approaches available in the literature to tackle Dynamic Vehicle Routing problems. The first approach relies on heuristic algorithms. In the second approach, called "competitive analysis approach," routing policies are designed to minimize the *worst-case* ratio between their performance and the performance of an optimal offline algorithm which has a priori knowledge of the entire input sequence. In the third approach, the routing problem is embedded within the framework of queueing theory. Routing policies are then designed to *stabilize* the system in terms of uniform boundedness of the number of outstanding tasks and to minimize typical queueing-theoretical cost functions such as the *expected time* the tasks remain in the queue. Since the generation of tasks and motion of the vehicles is within an Euclidean space, one can refer to this third approach as "spatial queueing theory".

Heuristic Approach

The main aspect of the heuristic approach is that routing algorithms are evaluated primarily via numerical, statistical and experimental studies, and formal performance guarantees are not available. A naïve, yet reasonable approach to design a heuristic algorithm for DVR would be to adapt classic queueing policies. However, perhaps surprisingly, this adaptation is not at all straightforward. For example, policies based on a First-Come First-Served discipline, whereby tasks are fulfilled in the order in which they arrive, are unable to stabilize the system for all stabilizable task arrival rates, in the sense that under such policies the average number of tasks grows over time without bound, even though there exist other policies that would maintain the number of tasks uniformly bounded (Bertsimas and van Ryzin [1991]).

The most widely applied approach is to combine static routing methods (e.g., VRP-like methods, nearest neighbor strategies, or genetic algorithms) and sequential re-optimization, where the re-optimization horizon is chosen heuristically. In particular, greedy nearest neighbor strategies, whose formal characterization still represents an open problem, are known to perform particularly well in some notable cases (Bertsimas and van Ryzin [1991]). However, the joint selection of a static routing method and of the re-optimization horizon in the presence of vehicle and task constraints (e.g., differential motion constraints, or task priorities) makes the application of this approach far from trivial. For example, one can show that an erroneous selection of the re-optimization horizon can lead to pathological scenarios where no task *ever* receives service (Pavone [2010]). Additionally, performance criteria in dynamic settings commonly differ from those of the corresponding static problems. For example, in a dynamic setting, the time needed to complete a task may be a more important factor than the total vehicle travel cost.

Competitive Analysis Approach

The distinctive feature of the competitive analysis approach is the method used to evaluate an algorithm's performance, which is called *competitive analysis*. In competitive analysis, the performance of a (causal) algorithm is compared to the performance of a corresponding offline algorithm (i.e., an algorithm that has *a priori* knowledge of the entire input) in the worst-case scenario. Specifically, an algorithm is c-competitive if its cost on *any* problem instance is at most c times the cost of an optimal offline algorithm:

$$\mathrm{Cost}_{\mathrm{causal}}(I) \leq c\ \mathrm{Cost}_{\mathrm{optimal\ offline}}(I), \qquad \textit{for all} \text{ problem instances } I.$$

In the recent past, several dynamic vehicle routing problems have been successfully studied in this framework, under the name of the online traveling repairman problem [Jaillet and Wagner, 2006], and many interesting insights have been obtained. However, the competitive analysis approach has some potential disadvantages. First, competitive analysis is a *worst-case* analysis, hence, the results are often overly pessimistic for normal problem instances, and potential statistical information about the problem (e.g., knowledge of the spatial distribution of future tasks) is often neglected. Second, the worst-case analysis usually requires a *finite* horizon problem formulation, which precludes the study of useful properties such as stability. Third, competitive analysis is used to bound the performance relative to an optimal offline algorithm, and thus it does not give an absolute measure of performance. In other words, with this approach one minimizes the "cost of causality" in the worst-case scenario, but not necessarily the minimum worst-case cost. Finally, many important real-world constraints for DVR, such as time windows, priorities, differential constraints on vehicle's motion and the requirement of teams to fulfill a task have so far proved to be too complex to be considered in the competitive analysis framework [Golden et al., 2008, page 206]. Some of these drawbacks have been recently addressed by Van Hentenryck et al. [2009] where a combined stochastic and competitive analysis approach is proposed for a general class of combinatorial optimization problems and is analyzed under some technical assumptions.

Spatial Queueing Theory

Spatial queueing theory embeds the dynamic vehicle routing problem within the framework of queueing theory. Spatial queueing theory consists of three main steps, namely development of a spatial queueing model, establishment of fundamental limitations of performance, and design of algorithms with performance guarantees. More specifically, the formulation of a model entails detailing four main aspects:

1. A model for the *dynamic* component of the environment: this is usually achieved by assuming that new events are generated (either adversarially or stochastically) by an exogenous process.
2. A model for targets/tasks: tasks are usually modeled as points in a physical environment distributed according to some (possibly unknown) distribution, might require a certain level of on-site service time, and can be subject to a variety of constraints, e.g., time windows, priorities, etc.
3. A model for the vehicles and their motion: besides their number, one needs to specify whether the vehicles are subject to algebraic (e.g., obstacles) or differential (e.g., minimum turning radius) constraints, sensing constraints, and fuel constraints. Also, the control could be centralized (i.e., coordinated by a central station) or decentralized, and subject to communication constraints.
4. Performance criterion: examples include the minimization of the waiting time before service, loss probabilities, expectation-variance analysis, etc.

Once the model is formulated, one seeks to characterize fundamental limitations of performance (in the form of lower bounds for the best achievable cost); the purpose of this step is essentially twofold: it allows the quantification of the degree of optimality of a routing algorithm and provides structural insights into the problem. As for the last step, the design of a routing algorithm usually relies on a careful combination of static routing methods with sequential re-optimization. Desirable properties for the static methods are: (i) the static problem can be solved (at least approximately) in polynomial time, and (ii) the static method is amenable to a statistical characterization (this is essential for the computation of performance bounds). Formal performance guarantees on a routing algorithm are then obtained by quantifying the ratio between an upper bound on the cost delivered by that algorithm and a lower bound for the best achievable cost. Such a ratio, being an estimate of the degree of optimality of the algorithm, should be close to one and possibly independent of system parameters. The proposed algorithms are finally evaluated via numerical, statistical and experimental studies, including Monte-Carlo comparisons with alternative approaches.

An interesting feature of this approach is that the performance analysis usually yields scaling laws for the quality of service in terms of model data, which can be used as useful guidelines to select system parameters when feasible (e.g., number of vehicles).

In order to make the model tractable, tasks are usually considered "statistically independent" and their arrival process is assumed stationary (with possibly unknown parameters). These assumptions, however, can be unrealistic in some scenarios, in which case the competitive analysis approach may represent a better alternative. Pioneering work in this context is that of Bertsimas and van Ryzin [1991], who introduced queueing methods to solve the baseline DVR problem (a vehicle moves along straight lines and visits tasks whose time of arrival, location and on-site service are stochastic; information about task location is communicated to the vehicle upon task arrival). Next section provides an overview of the application of spatial queueing theory to such simplified DVR problem, referred to in the literature as Dynamic Traveling Repairman Problem (DTRP).

Applying Spatial Queueing Theory to DVR Problems

Spatial Queueing Theory Workflow for DTRP

Model

The DTRP, which, incidentally, captures well the salient features of the UAV scenario outlined in the Motivation Section, can be modeled as follows. In a geographical region $\mathcal{Q}$ of area A, a dynamic process generates spatially localized tasks. The process generating tasks is modeled as a spatio-temporal Poisson process, i.e., (i) the time between each pair of consecutive task generation has an exponential distribution with intensity $\lambda > 0$ and (ii) upon arrival, the locations of tasks are independently and uniformly distributed in $\mathcal{Q}$. The location of the new tasks is assumed to be immediately available to a team of m servicing vehicles. The vehicles provide service in $\mathcal{Q}$, traveling at a speed at most equal to v; the vehicles are assumed to have unlimited fuel and task-servicing capabilities. Each task requires an independent and identically distributed amount of on-site service with finite mean duration $\bar{s} > 0$. A task is completed when one of the vehicles moves to its location and performs its on-site service. The objective is to design a *routing policy* that maximizes the quality of service delivered by the vehicles in terms of the average steady-state time delay $\overline{T}$ between the generation of a task and the time it is completed (in general, in a dynamic setting, the focus is on the quality of service as perceived by the "end user," rather than, for example, fuel economies achieved by the vehicles). Other quantities of interest are the average number $\overline{N}$ of tasks waiting to be completed and the waiting time $\overline{W}$ of a task before its location is reached by a vehicle. These quantities, however, are related according to $\overline{T} = \overline{W} + \bar{s}$ (by definition) and by Little's law, stating that $\overline{N} = \lambda \overline{W}$, for stable queues.

The system is considered stable if the expected number of waiting tasks is uniformly bounded at all times, or equivalently, that tasks are removed from the system at least at the same rate at which they are generated. In the case at hand, the time to complete a task is the sum of the time to reach its location (which depends on the routing policy) plus the time spent at that location in on-site service (which is independent of the routing policy). Since, by definition, the service time is no shorter than the on-site service time $\bar{s}$, then a weaker necessary condition for stability is $\varrho := \lambda \bar{s}/m < 1$; the quantity ϱ measures the fraction of time the vehicles are performing on-site service. Remarkably, it turns out that this is also a sufficient condition for stability; note that this stability condition is independent of the size and shape of $\mathcal{Q}$, and of the speed of the vehicles.

Fundamental limitations of performance

To derive lower bounds, the main difficulty consists in bounding (possibly in a statistical sense) the amount of time spent to reach a target location. The derivation of these bounds becomes simpler in asymptotic regimes, i.e., looking at cases when $\varrho \to 0^+$ and $\varrho \to 1^-$, which are often called "light load" and "heavy load" conditions, respectively.

Consider first the case in which $\varrho \to 0^+$ (light load regime). A set of m points is called the m-median of $\mathcal{Q}$ if it globally minimizes the expected distance between a random point sampled uniformly from $\mathcal{Q}$ and the closest point in such set. In other words, the m-median of $\mathcal{Q}$ globally minimizes the function

$$H_m(p_1, p_2, \ldots, p_m) := \mathbb{E}\left[\min_{k \in \{1,\ldots,m\}} \|p_k - q\|\right] = \frac{1}{A} \int_{\mathcal{Q}} \min_{k \in \{1,\ldots,m\}} \|p_k - q\| \, dq.$$

Let H_m^* be the global minimum of this function. Geometric considerations show that H_m^* scales proportionally to $\sqrt{A/m}$.

Incidentally, the m-median of $\mathcal{Q}$ induces a Voronoi partition that is called *Median Voronoi Tessellation*, whose importance will become clear in the next section. Recall that the Voronoi Diagram of $\mathcal{Q}$ induced by points $(p_1, \dots, p_m)$ is defined by

$$V_i = \left\{ q \in \mathcal{Q} | \, \|q - p_i\| \leq \|q - p_j\|, \forall j \neq i, \; j \in \{1, \dots, m\} \right\},$$

where V_i is the region associated with the i-th "generator" point. The distance H_m^* certainly provides a lower bound on the expected distance traveled by a vehicle to reach a task, and hence one obtains the lower bound

$$\overline{T} \geq \frac{H_m^*}{v} + \bar{s}.$$

This lower bound is tight in light load conditions ($\varrho \to 0^+$), as it will be seen in the next section.

Consider now the case in which $\varrho \to 1^-$ (heavy load). Let $\overline{D}$ be the average travel distance per task for some routing policy. By using arguments from geometrical probability (independent of algorithms), one can show that $\overline{D} \geq \beta_2 \sqrt{A}/\sqrt{2\overline{N}}$ as $\varrho \to 1^-$, where β_2 is a constant that will be specified later. As discussed, for stability, one needs $\bar{s} + \overline{D}/v < m/\lambda$. Combining the stability condition with the bound on the average travel distance per task, one obtains

$$\bar{s} + \frac{\beta_2 \sqrt{A}}{v\sqrt{2\overline{N}}} \leq \frac{m}{\lambda}.$$

Since, by Little's law, $\overline{N} = \lambda \overline{W}$ and $\overline{T} = \overline{W} + \bar{s}$, one finally obtains (recall that $\varrho = \lambda \bar{s}/m$):

$$\overline{T} \geq \frac{\beta_2^2}{2} \frac{A}{v^2} \frac{\lambda}{m^2 (1 - \varrho)^2} + \bar{s}, \qquad (\text{as } \varrho \to 1^-).$$

A salient feature of the above lower bound is that it scales *quadratically* with the number of vehicles (as opposed to the square-root scaling law one has in light load conditions); note, however, that congestion effects are not included in this model. This bound also shows that the quality of service, which is proportional to $1/(1 - \varrho)^2$, degrades much faster as the target load increases than in non-spatial queueing systems (where the growth rate is proportional to $1/(1 - \varrho)$).

Design of routing algorithms

The design of an optimal light-load policy essentially relies on mimicking the proof strategy employed for the light-load lower bound. Specifically, a routing policy whereby 1) one vehicle is assigned to each of the m median locations of $\mathcal{Q}$, 2) new tasks are assigned to the nearest median location and its corresponding vehicle, and 3) each vehicle services tasks according to a First-Come-First-Served policy is asymptotically optimal, i.e.,

$$\overline{T} \to \frac{H_m^*}{v} + \bar{s}, \qquad (\text{as } \varrho \to 0^+).$$

Note that under this strategy "regions of dominance" are implicitly assigned to vehicles according to a Median Voronoi Tessellation.

The heavy-load case is more challenging. Consider, first, the following single-vehicle routing policy, based on a partition of $\mathcal{Q}$ into $p \geq 1$ sub-regions $\{\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_p\}$ of equal area A/p. Such a partition can be obtained, e.g., as sectors centered at the median of $\mathcal{Q}$. Define a cyclic ordering for the sub-region, such that, e.g., if the vehicle is in region $\mathcal{Q}_i$ the "next" region is $\mathcal{Q}_j$, where j follows i in the cyclic ordering (in other words, $j = (i+1) \bmod p$).

1. If there are no outstanding tasks, move to the median of the region $\mathcal{Q}$.
2. Otherwise, visit the "next" sub-region; subregions with no tasks are skipped. Compute a minimum-length path from the vehicle's current position through all the outstanding tasks in that subregion. Complete all tasks on this path, ignoring new tasks generated in the meantime. Repeat.

The problem of computing the shortest path through a number of points is related to the well-known Traveling Salesman Problem (TSP). While the TSP is a prototypically hard combinatorial optimization problem, it is well known that the Euclidean version of the problem can be approximated efficiently [**?**]. Furthermore, the length ETSP(n) of a Euclidean TSP through n points independently and uniformly sampled in $\mathcal{Q}$ is known to satisfy the following property:

$$\lim_{n\to\infty} \mathrm{ETSP}(n)/\sqrt{n} = \beta_2 \cdot \sqrt{A}, \qquad \text{almost surely,}$$

where $\beta_2 \approx 0.712$ is a constant (the same β_2 constant that appeared in the previous section) [**?**].

It can be shown that, using the above routing policy, the average system time $\overline{T}$ satisfies

$$\overline{T} \leq \gamma(p)\frac{A}{v^2}\frac{\lambda}{(1-\varrho)^2} + \bar{s}, \qquad (\text{as } \varrho \to 1^-),$$

where $\gamma(1) = \beta_2^2$, and $\gamma(p) \to \beta_2^2/2$ for large p. These results critically exploit the statistical characterization of the length of an optimal TSP tour. Hence, the proposed policy achieves a quality of service that is arbitrarily close to the optimal one, in the asymptotic regime of heavy load (and, indeed, also of light load).

The above single-vehicle routing policies can be fairly easily lifted to an efficient multi-vehicle routing policy. The key idea (akin to the one in the light load case) is to 1) partition the workspace into m *regions of dominance* (with disjoint interiors and whose union is $\mathcal{Q}$), 2) assign one vehicle to each region, and 3) have each vehicle follow a single-vehicle routing policy within its own region. This approach leads to the following multi-vehicle routing policy for the DTRP problem:

1. Partition $\mathcal{Q}$ into m regions of dominance of *equal area* and assign one vehicle to each region.
2. Each vehicle executes a single-vehicle DTRP policy in its own subregion.

Using as single-vehicle policy the routing policy described above, the average system time $\overline{T}$ in heavy-load satisfies

$$\overline{T} \leq \gamma(p)\,\frac{A}{v^2}\,\frac{\lambda}{m^2\,(1-\varrho)^2} + \bar{s}, \qquad (\varrho \to 1^-).$$

Hence, by comparing this result with the corresponding lower bound, one concludes that a simple partitioning strategy leads to a multi-vehicle routing policy whose performance is arbitrarily close to the optimal one in heavy load.

Mode of implementation

The scalability of the control policies to large groups of vehicles often requires a distributed implementation of multi-vehicle routing strategies. For the DTRP, a distributed implementation can be obtained by devising *decentralized algorithms for environment partitioning*. In the solution proposed in Pavone [2010], power diagrams are the key geometric concept to obtain, in a decentralized fashion, partitions suitable for both the light load case (requiring, as seen before, a Median Voronoi Tessellation) and the heavy load case (requiring an equal-area partition). The power diagram of $\mathcal{Q}$ is defined as

$$V_i = \Big\{ q \in \mathcal{Q} | \, \|q - p_i\|^2 - w_i \leq \|q - p_j\|^2 - w_j, \, \forall j \neq i, \; j \in \{1, \dots, m\} \Big\}.$$

where $(p_i, w_i) \in \mathcal{Q} \times \mathbb{R}$ are a set of "power points", and V_i is the subregion associated with the i-th "generator" power point. Note that power diagrams are a generalization of Voronoi diagrams: when all weights are equal, the power diagram and the Voronoi diagram are identical. The basic idea, then, is to associate to each vehicle i a *virtual* power point, which is an artificial (or logical) variable whose value is locally controlled by the i-th vehicle. The cell V_i becomes the region of dominance for vehicle i, and each vehicle updates its own power point according to a *decentralized* gradient-descent law with respect to a coverage function (see chapter on Optimal Deployment and Spatial Coverage), until the desired partition is achieved. The reader is referred to Pavone [2010] for more details.

Extensions and Discussion

By integrating additional ideas from dynamics, teaming, and distributed algorithms, the spatial queueing theory approach has been recently applied to scenarios with complex models for the tasks such as time constraints, service priorities, translating tasks, and adversarial generation, has been extended to address aspects concerning robotic implementation such as complex vehicle dynamics, limited sensing range, and team forming, and has even been tailored to integrate humans in the design space, see Bullo et al. [2011] and references therein. Despite the significant modeling differences, the "workflow" is essentially the same as in the DTRP: a queueing model that captures the salient features of the problem at hand, characterization of the fundamental limitations of performance, and design of algorithms with provable performance bounds. The last step, as for the DTRP, often involves lifting a single-vehicle policy to a multi-vehicle policy through the strategy of environment partitioning. Within this context, a number of partitioning schemes and corresponding *decentralized* partitioning algorithms relevant to a large variety of DVR problems are discussed in Pavone et al. [2009].

This workflow efficiently and transparently *decouples* the three decision-making problems mentioned in the Introduction Section, i.e., "task allocation", "service scheduling", and "loitering paths". In fact, task allocation is addressed via the strategy of environment partitioning, service scheduling is address by applying a single-vehicle routing policy within the individual regions of dominance, and the loitering paths resolve in placing the vehicles at or around specific points within the dominance regions (e.g., the median). Note, however, that in some important cases, e.g., DVR problems where goods have to be transported from a pick-up location to a delivery location, or where vehicles are differentially constrained and operate in a "congested" workspace, multi-vehicle policies that rely on static partitions perform poorly or are not even feasible (Pavone et al. [2009]), and task allocation and service scheduling need to be addressed as tightly coupled.

Through spatial queueing theory one is usually able to characterize the performance of multi-vehicle routing policies in asymptotic regimes. To ensure "satisfactory" performance under general operation conditions, a common strategy is to consider heuristic modifications to a baseline asymptotically-efficient routing policy in such a way that, on the one hand, asymptotic performance is preserved, and, on the other hand, light- and heavy-load performances are “smoothly” and efficiently blended in the intermediate load case. The interested reader can find more information in Bullo et al. [2011].

Finally, from a technical standpoint, one should note that spatial queueing models are *inherently* different from traditional, non-spatial queueing models. The main reason is that in spatial queueing models the “service time” per task has both a *travel* and an *on-site* component. Although the on-site service requirements can be modeled as “statistically” independent, the travel times are inherently statistically coupled. Hence, in contrast to standard queueing models, service times in spatial queueing models are statistical dependent, and this deeply affects the solution to the problem, as noted above.

Cross References

- Averaging algorithms and consensus
- Control of networked systems, Overview
- Flocking in Control of Networked Systems
- Optimal deployment and spatial coverage
- Steering laws for interacting particles

Recommended Reading

D. J. Bertsimas and G. J. van Ryzin. A stochastic and dynamic vehicle routing problem in the Euclidean plane. *Operations Research*, 39:601–615, 1991.

F. Bullo, E. Frazzoli, M. Pavone, K. Savla, and S.L. Smith. Dynamic vehicle routing for robotic systems. *Proceedings of the IEEE*, 99(9):1482 –1504, 2011.

B. Golden, S. Raghavan, and E. Wasil. *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *Operations Research/Computer Science Interfaces*. Springer, 2008.

P. Jaillet and M. R. Wagner. Online routing problems: Value of advanced information and improved competitive ratios. *Transportation Science*, 40(2):200–210, 2006.

M. Pavone. *Dynamic Vehicle Routing for Robotic Networks*. PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, June 2010.

M. Pavone, K. Savla, and E. Frazzoli. Sharing the load. *IEEE Robotics and Automation Magazine*, 16(2): 52–61, 2009.

P. Toth and D. Vigo, editors. *The Vehicle Routing Problem*. Monographs on Discrete Mathematics and Applications. SIAM, 2001. ISBN 0898715792.

P. Van Hentenryck, R. Bent, and E. Upfal. Online stochastic optimization under time constraints. *Annals of Operations Research*, 177(1):151–183, 2009.