Chapter 2

Dynamic Transportation Problems

This chapter begins with a general characterization of dynamic problems, specifically pointing out the differences between dynamic and static problems and proposing a configuration framework for dynamic algorithms (Section 2.1). Subsequently, the most relevant dynamic real-life applications are elaborated and connected with the associated theoretical problem definitions (Section 2.2). Finally, two dynamic problem definitions that have been chosen for further investigation in Chapters 4 and 5 are introduced (Section 2.3).

2.1 Characteristics of Dynamic Problems

As a preliminary matter, the **term "dynamic"** (also referred to as "real time" or "on-line") needs to be defined. One of the earliest references related to Vehicle Routing is Psaraftis (1988):

> *"In a dynamic vehicle routing problem, inputs may (and generally will) change (or be updated) during the execution of the algorithm and the eventual execution of the route. Algorithm execution and route execution are processes that evolve concurrently in a dynamic situation, in contrast to a static situation in which the former process clearly precedes (and has no overlap with) the latter."*

In a definition by Pankratz (2005), the need for "irreversible" decisions under incomplete information is emphasized:

> *"planning and execution are overlapping processes, and planning decisions, which may be irreversible, have to be taken before all problem data become known."*

A dispatcher is therefore forced to plan in a sequential or **rolling horizon** manner. He solves a part of the overall problem on the basis of the information available at the present moment. This partial problem is denoted as **static subproblem**. When new information arrives the static subproblem changes and the dispatcher has to resolve the problem.

The static subproblem is not only affected by newly occurring information, but also by **fixation**. Fixation is defined as the successive alteration of provisional planning decisions into permanent ones. It is basically triggered by proceeding time, i.e., if an event's scheduled execution time is reached, the event is permanently fixed. A fixation reduces

Steffen Schorpp, *Dynamic Fleet Management for International Truck Transportation*,
DOI 10.1007/978-3-8349-6675-9_2,

the available planning options and therefore is a second source of changes to the static subproblem.

The static subproblem's **planning horizon** depends on the applied solution approach: it ranges from the present moment up to the last scheduled activity. All decisions are of provisional type and can be changed by an improvement procedure, unless they are finally fixed.

There are several **sources** inducing dynamism of a vehicle routing problem: New customer orders, cancelation or modification of already known customer orders, revealment or changes of actual demand level and customer service time, changes in vehicle travel time (due to unforeseen events, such as traffic jams or unexpected delays), up to complete vehicle breakdown.

Differences: Dynamic vs. Static

To identify the specific characteristics of a dynamic problem, a comparison with a classic static problem is useful. A whole catalog of such differences was defined by Psaraftis (1988) and is subsequently quoted in a shortened form:

- **Time dimension is essential:** In static vehicle routing, time dimension may or may not be an important factor in the problem. If there is a scheduling component alongside the routing component, time dimension is essential. Actually, most classic generic routing problems, such as Traveling Salesman Problem (TSP) or Vehicle Routing Problem (VRP), do not have a scheduling component. In contrast, time dimension is essential in every dynamic vehicle routing situation, whether it is time constrained or not. It is necessary to keep track of how vehicle schedules and scheduling options evolve dynamically over time.

- **Problem may be open-ended:** In a static situation, the duration of the routing process is more or less bounded or known in advance. The duration of such a process in a dynamic situation may neither be bounded nor known.

- **Future information may be imprecise or unknown:** In a static context, information about all problem inputs is assumed to be of the same quality, irrespective of where within the schedule this input happens to be (beginning, middle, or end). This is not the case in a dynamic problem, in which information on any input is usually precise for events that happen in real time, but more tentative for events that may occur in the future. Probabilistic information about the future may be available.

- **Near-term events are more important:** Because of uniformity of information quality and lack of input updates, all events (whether in the beginning, in the middle, or at the end of a vehicle's route) carry the same "weight" in a static context. In dynamic routing, it would be unwise to immediately commit vehicle resources to requirements that will have to be met in the distant future. This is because other intermediate events may render such decisions suboptimal, and because such future information may change anyway. Focusing on near term events is therefore an essential aspect of dynamic vehicle routing.

- **Information update mechanisms are essential:** Virtually all inputs into a dynamic routing problem are subject to revision at any time during the execution of the route. Therefore, update mechanisms are an integral part of the algorithmic structure. Data structure and database management techniques that help revise problem inputs efficiently, as well as adeptly figure out the consequences of such revisions, are central to a dynamic routing scheme. In contrast, in a static scenario, such mechanisms are not necessary.

- **Resequencing and reassignment decisions may be warranted:** In a dynamic vehicle routing situation, the appearance of a new input may render decisions that have already been made prior to that input's appearance suboptimal. This fact concerns both sequencing and assignment decisions. Thus, the appearance of new input may necessitate either the resequencing of the stops of one (or more) vehicle(s), or the reassignment of those vehicles to demands requesting service (or both).

- **Fast computation times are necessary:** The need to reoptimize routes and/or vehicle assignments on a continual basis in real time necessitates computation times faster than those necessary in a static situation. In a static situation, computation runs may take several hours or overnight. In a dynamic routing situation, if new information is available, the dispatcher wishes to know the solution to a particular problem as soon as possible (within minutes or seconds).

- **Indefinite postponement mechanisms are essential:** Indefinite deferment means that the service of a particular demand can be postponed indefinitely due to that demand's unfavourable geographical characteristics relative to other demands. The problem can be handled by introducing time constraints.

- **Time constraints may be different:** Dynamic routing inputs, such as Earliest Pickup Time (EPT) and Latest Pickup Time (LPT), tend to be softer than in a static situation. If a "hard" deadline makes a routing problem infeasible, it is far better to renegotiate that deadline so as to make it feasible than it is to declare infeasibility and quit.

- **Flexibility to vary vehicle fleet size is lower:** In theory, another alternative to denying service to a customer, if a time constraint cannot be met, is to add an additional vehicle, at a cost, to serve that customer. However, this proposition may not necessarily be viable in a dynamic vehicle routing because it may not be possible to have access to backup vehicle resources in real time. In a static situation, the time gap between execution of the algorithm and execution of the route is usually long enough to allow for such a decision to be made.

These aspects are complemented by a comparison of the *open order characteristic* in a static and in a dynamic environment (cp. Sandvoß, 2002). In the static context, all orders are initially known, being successively completed as time proceeds. The number of unfulfilled orders decreases monotonously. Figure 2.1 shows an exemplary curve.

In a dynamic problem, however, the initial order level is lower (or non-existent), since dynamic orders are revealed later (up to time T). This may result in time intervals of an increasing number of open orders when the number of new occurring orders exceeds the number of completed orders. After the latest Call-In time T, the number of unfulfilled orders monotonously decreases like in the static case (cp. Figure 2.1).

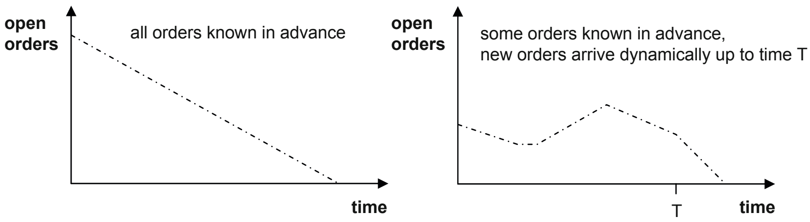

Figure 2.1: Open order characteristics for static and dynamic problems

Degrees of Information Availability and Possible Reactivity

In the following, four problem categories are differentiated, based on *initial information availability and information certainty* as well as *possible reactivity*:

- **static**,
- **static stochastic**,
- **dynamic stochastic**, and
- **dynamic**.

Initial information availability is related to the "quantity of information" that is known to the decision maker before the planning horizon starts. We use the word "complete" to denote the condition of having information about all relevant facts available (e.g. about all customers and the associated locations, all specific demand levels, etc.). In contrast, the word "incomplete" is used for the situation of initially not having all basic information ready. Thus, some customers, demands, etc. may be revealed during the planning horizon.

However, the availability of the initial information predicates nothing about initially and subsequently revealed *information's certainty*: We call information "certain" if it is given as an exact deterministic value. Otherwise, the information is subject to "uncertainty", e.g. when a probability distribution is given instead of an exact value, or if the exact value may be subject to further changes.

Figure 2.2 shows the associated characteristics of *initial information and information certainty* for the regarded problem categories (column one). The second column describes the possible availability of stochastic information about the future. To the right, some bars are plotted in order to symbolize the relative "degree of availability and certainty of information". The size of the bars is related to column one and two, and drops from the first to the fourth problem category, indicating lowest information availability in category four.

Afterwards a third column is given that states whether the associated problem category allows for "replanning" during plan execution or not. The bars to the right visualize the relative extent of *possible reactivity* and show increased reactivity for categories three and

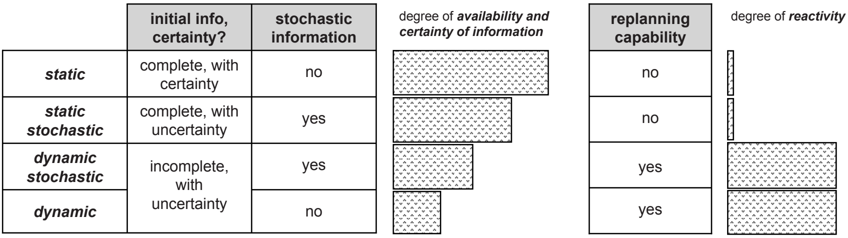

Figure 2.2: Information availability and replanning options for different planning categories

four, compared to categories one and two, which do not allow for replanning at all.

In the first category, **static**, all information is "completely" available before the planning horizon starts and is not subject to any changes ("certainty"). Stochastic information of the future and a re-planning option are neither available nor required. This case corresponds to the classical static planning situation, hence several traditional procedures may be applied in order to find the best solution.

In the second category, **static stochastic**, initial information is "complete" but subject to "uncertainty". Parts of information are given as discrete or continuous random variables, for example, approximated through the use of historical data. The static character requires an *initial planning* run to construct an "a-priori" solution that optimizes the expected value of a given objective function (first stage). All possible decisions are fixed, including some strategies on "how to react to unexpected developments". Later changes of these recourse strategies are not possible ("no re-planning").

When the routes are actually executed in the second stage, these initially specified strategies (*recourse actions*) are applied to the first stage solution in order to address the current realization (e.g., "skip a customer who does not show up", or "send the vehicle back to depot before it resumes its tour because of capacity shortage"). Typical representatives are the "Probabilistic TSP", "VRP with stochastic customers", "VRP with stochastic demands", and "VRP with stochastic customers and demands". (cp. Ichoua et al., 2006)

The third category is denoted as **dynamic stochastic**. Initial information is "no longer complete". Instead, some information is revealed during the planning horizon. In addition, there is "no certainty" about the given information, which may be subject to repeated changes. As in the second category, some stochastic information on the future is available, partially compensating for the lack of basic information quantity. There is no longer the need to decide on reaction strategies "a-priori", instead, an appropriate online replanning may be performed in a rolling horizon manner.

In order to deal with the stochastic information, several approaches can be chosen. In a *sampling approach*, for example, the algorithm generates a sufficient number of future scenarios (by drawing from the given probability distributions) and uses the scenarios to find a good and robust solution at each rolling horizon step. In contrast thereto, a *stochastic algorithm* explicitly incorporates the current information and probabilities of future events into its objective function: The given probabilities are used for calculation of "recourse functions", which include the costs that the assumed solution scenario does not occur. Other approaches *directly derive some measures*, like "vehicle re-allocation" or "scheduling of extra waiting times in promising regions" when certain probabilities exceed some threshold.

In the fourth category, **dynamic**, initial information (if available at all) is "incomplete". All initial and later revealed information is subject to changes ("uncertainty"). Information about the future is not available. However, there is a *replanning capability* available for appropriate reaction to dynamic information. In comparison to the previous categories, the degree of initial information availability and certainty is the lowest. However, especially in comparison with the static stochastic situation, this lack of information may be compensated by the dynamic replanning capability. A dynamic planning situation is usually handled with a rolling horizon planning approach, which includes the new obtained information step-by-step.

Measuring Dynamism

In the previous explanations, several sources that may cause dynamism were mentioned, especially dynamic requests and dynamic travel times. According to Larsen (2000), a measure for dynamism shall quantify the extent of new information emerging during the operational phase of the system, thus being helpful for evaluating and comparing the "difficulty" of various problem instances.

A first "request-related" measure was proposed by Lund et al. (1996) and Larsen (2000). They define the *basic degree of dynamism (dod)* as the number of dynamic requests n_{dyn} relative to the total number of requests n.

$$dod = \frac{n_{dyn}}{n}$$

Larsen et al. (2002) distinguish three levels of dynamic systems, based on the degree of dynamism. First, *weakly dynamic systems*, with up to 20% dynamic orders, where reaction time is considerably longer compared to other dynamic problems (e.g. distribution of heating oil to private households, residential cable and telephone repair services, or the transportation of the elderly and physically disabled). The second group consists of *moderately dynamic systems*, having a substantial part of dynamic requests (20% up to 80%). As typical applications here, overnight mail services and appliance repair are mentioned, where scheduled customers are interspersed with dynamic ones that need immediate attention due to the gravity of their request. Finally, there are *strongly dynamic systems* with more than 80% of dynamically occurring customers, where frequent changes in data have to be handled within minimal response times (e.g. taxi and emergency services).

The same authors have also formulated extended versions of the "degree of dynamism". The first extension additionally incorporates the relative position of the Call-In time $t_i \in (0 < t_i \leq T)$ of the dynamic order i, in relation to the latest possible Call-In time T.

The resulting measure is called *effective degree of dynamism (edod)*:

$$edod = \frac{1}{n} \sum_{i=1}^{n_{dyn}} \frac{t_i}{T}$$

Further refinement is achieved by considering the temporal gap between an order's Call-In time and its time window. The time window of request i is specified by earliest service time e_i and latest service time l_i. The time gap between t_i and l_i is called *reaction time* and is used to calculate the measure *edod-tw*:

$$edod - tw = \frac{1}{n} \sum_{i=1}^{n} \frac{T - (l_i - t_i)}{T}$$

Figure 2.3 visualizes the concept of *edod-tw*: t_1 and t_2 symbolize different Call-In times with regard to the same time window. Since $t_2 > t_1$, there is a shorter reaction time in the second case.

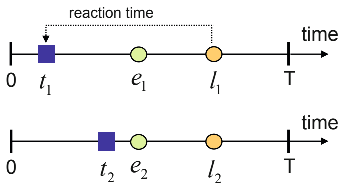

Figure 2.3: Effective degree of dynamism - with time windows

The work of Tjokroamidjojo et al. (2006) investigates the impact of different *reaction times* for a Single Load Pickup and Delivery Problem with Time Windows (SLPDPTW). It is shown for a theoretical test environment that the use of advance load information (with longer reaction time) produces significantly better results compared to a situation with Call-In at the latest possible time. A mandatory assumption for such a statement is an adequate replanning option in order to use the available "reaction time". The authors show that the absence of such an option may completely invalidate the advantage of "early information".

Due to the fact that dynamic requests can be characterized as the most important source of dynamism, these specific measures have their warranty. On the other hand, further possible sources of uncertainty are completely neglected. An approach which tries to cover other sources of dynamism is proposed by Schumann et al. (2009). Unfortunately, it is rather a theoretical type of consideration.

The authors extend Larsen's *edod-tw* formula. Instead of all orders n, they just consider dynamic orders n_{dyn} and propose the inclusion of additional dynamic events. The authors select solely dynamic events $\gamma \in \Gamma$ that may invalidate the current plan or may decrease solution quality. The set Γ includes (i) decrease of l_i, (ii) increase of e_i, (iii) increase of travel time between two nodes, (iv) increase of a customer's demand and (v) occurrence of new requests. The proposed dynamic measure ϕ is calculated with the formula:

$$\phi = \frac{1}{|\Gamma|} \sum_{\gamma}^{\Gamma} \frac{T - (h_\gamma - t_\gamma)}{T}$$

The value of h_γ depends on the specific event: (i) if l_i is decreased, h_γ is chosen to be the new value of l_i; (ii) if e_i is increased, and (iii) if a link's travel time increases, the first customer in the tour, whose time window cannot be met any longer, is identified. Then h_γ is chosen to be this customer's latest service time; (iv) if a customer's demand increases, the first subsequent customer in the tour is identified, who can no longer be served because of limited capacity. Then h_γ is chosen to be the departure time from this node's predecessor; (v) if a new order occurs, h_γ is chosen to be the latest service time associated with the new order.

If in cases (ii), (iii) and (iv), no customer satisfying the query can be found, the formula's numerator is set to zero for the considered γ-event.

A drawback of the proposed measure is its dependency on the current plan. An identical dynamic data set may result in different levels of dynamism ϕ, depending on the performance of the underlying planning approach. In addition, "positive events", e.g. reduction in travel times, are completely neglected, even though such "positive events" also have to be handled, causing "dynamic stress" to the planning system.

In general, a meaningful measure for dynamism should incorporate three main aspects:

- the type of new information and the magnitude of changes,
- the available reaction time for the planning system between disclosure date and possibly resulting negative impact, and
- the number of changes made known to the planning system per time unit ("stress").

In addition, it should be certainly independent of the applied solution procedure.

Due to the complexity of this task, to our knowledge no such measure has been developed to date.

Configuration Framework for Dynamic Algorithms

The most important and distinctive feature of a dynamic algorithm is the capability to **quickly** produce a new **good plan** after new information has arrived. The way to perform this task can be described with the aspects of the following configuration framework (cp. Bock, 2004).

1. Technique of Adjustment

This aspect answers the question of what kind of **methodological approach** to use **to incorporate new information**. There are easy *predefined decision rules* like "Nearest Neighbor", or more advanced *heuristic approaches*, which construct a feasible plan and try to improve it by applying exchange operators (possibly guided by a metaheuristic). In addition, *exact approaches* (e.g. Column Generation based) may also be applied, either in their original "exact" setting (for small instances) or in a heuristical way (for greater instances).

Intuitively (from a static perspective), the use of a more advanced approach applied to the static subproblems should result in a better overall performance. Interestingly, this point is subject to discussion. While Yang et al. (2004) report "that fully re-optimizing each time (seeking the optimal solution) leads to overall better performance under various testing situations", Hvattum et al. (2006) experience that "a better solution to the static subproblem does not necessarily lead to a better overall solution".

In fact, in a few cases we had the same "non intuitive" experience in our tests (see Section 4.5.1). This behavior can be explained as follows. As we saw in the definition of a dynamic problem, it requires planning decisions under incomplete information, which are to some extent irreversible. Even if a "more advanced" procedure A produces a better solution than an inferior procedure B under information of time t, new information at time t+1 can completely render the situation. Perhaps procedure A has fixed some decisions (optimal at time t), which emerge sub-optimal under new information of time t+1. So, in total, it is possible that procedure A, despite its structural dominance, is no more capable of catching up with procedure B in the remaining time.

However, in the most cases we made the "intuitive" experience that more advanced procedures applied to the static subproblems also produced better overall results.

2. Reaction of Adjustment

If former planning runs have been performed, there will be an obsolete solution available (not including the newly arrived information). This second categorization aspect decides **how to use** such **old planning solutions**. One approach is to completely neglect former planning results and to build up a new feasible solution *from scratch*, incorporating all new information, treating it as equal to old information. In contrast thereto, a *constructive* approach takes the planning results from the last planning run as given and just updates it with all new information now available.

3. Frequency of Adjustment

This categorization answers the question of **how often the plan in execution is updated**, for example because of new information or improvements. Four different options can be distinguished. The first is that new information becoming available immediately triggers a new planning run and an update of the plan in execution (*event-based*). In the second option, new information is gathered over a specified period and if time is elapsed, a new planning run is started (*time driven*).

In the third option, new information is gathered up to a point where a specified number of new events has occurred, then a new planning run is started (*size driven*). The fourth option allows updates of the plan in execution, not only at every new event, but also when an improvement procedure, running simultaneously to execution, finds a better specification for the plan in execution (*continuous*).

4. Duration of Adjustment

Duration of Adjustment describes the **time available to the algorithm** for incorporating new information and for performing improvement. In a *time limit based* environment, a known finite *anticipation horizon* is available to incorporate new information. An improvement strategy can be developed so as to use the complete time available for a good search strategy (diversification and intensification; see Rochat and Taillard, 1995). Finding a value that achieves a good trade-off between execution time and solution quality is a challenging task: If the *anticipation horizon* is too small, there is not enough time available for the optimization procedure. Situations may be incorrectly assessed, resulting in inferior decisions. Otherwise, if the anticipation horizon is too long, planning opportunities may be lost due to the delayed possibility for reaction (see Ichoua et al., 2000).

In an *event-based* environment, duration is stochastic and not known to the algorithm. A feasible solution needs to be produced very fast, so that it is ready in case of the occurrence of the next event. Here, intensification strategies are especially appropriate for improvement. A further duration category is called *zero-time*. In this case, the algorithm just gets a very small amount of time (near zero) to incorporate new information into the plan in execution. Improvement is not performed.

5. Synchronization of Adjustment

In a real time planning environment, calculations of a control algorithm have to be done parallel to the progress of time in the real world. Several options are available to achieve **synchronization between algorithmic calculations and plan in execution**. The first (*prioritization of computation*) implies a stop of the execution of the current plan while calculations are performed. Afterwards, the plan in execution is updated with the new results and execution is continued. For practical reasons, this option seems to be hardly applicable.

In the second option (*extensive simultaneity*), execution and computation run simultaneously. The algorithm is allowed to change all decisions which are not due within a short *anticipation horizon*. A third approach specifies a larger part of the "plan in execution", which is not allowed to be changed (*prioritization of execution*), avoiding stress of frequent re-scheduling, but also reducing the potential solution space.

6. Scope of Adjustment

The scope describes how the algorithm is allowed to change decisions which become effective in the future. The scope is called *restricted* if some adjustments, which were technically possible, are not allowed. Examples are the restriction of diversion (not to allow a change of the destination when a vehicle is already traveling to a specified location) or of transshipment (not to allow the exchange of an already picked up load from one truck to another). The scope of adjustment is called *complete* if no future decision is restricted. However, in the most cases, the scope will be somehow *restricted*.

Simulation Techniques for Dynamic Algorithms

For evaluation of a new dynamic procedure's performance, simulation runs with various test data sets are usually applied. The best simulation mode depends on the individual algorithmic concept. For subsequent explanation, the simulation speed s is defined, meaning that 1 *hour* of real-life operations is simulated in $1/s$ *hours* of computer time.

If the algorithm possesses an improvement component, it may be beneficial to run the simulation in "real time" ($s = 1$), allowing for the same number of potential improvement operations as in reality. Here, higher simulation speeds ($s > 1$, with shorter simulation run time) result on average in an overall worse solution quality. Examples of real time simulation can be found in Shieh and May (1998), Gendreau et al. (2006) and Chen and Xu (2006). For a detailed investigation of simulation speed effects, see Section 5.5.

If there is no improvement component available, it is not necessary to choose the time consuming "real time" simulation (the time between two subsequent events would be wasted). Instead, it is sufficient to run an "event driven" simulation. In this case, it is not possible to give an exact value of simulation speed. It is only possible to estimate the number of events, multiply it with the average calculation time per event and set it into relation to the total simulated real time. Examples of event driven simulation can be found in Bent and van Hentenryck (2004), Fleischmann et al. (2004) and Tang and Hu (2005).

Performance Analysis for dynamic algorithms

The performance of a dynamic heuristic on a given dynamic test instance can be evaluated in five possible ways:

- by analytical derivation of the heuristic's worst case performance, compared to the optimal solution, obtained for the corresponding *static* instance,
- by comparison with the *optimal* solution obtained for the corresponding *static* instance,
- by comparison with a *heuristic* solution obtained for the corresponding *static* instance,
- by comparison with solutions achieved by other well-established dynamic heuristics for the original *dynamic* instance, and
- by comparison with *manually* achieved solutions of human planners, in case of a *dynamic* real-life instance.

The value of the **first case** is more of a theoretical nature; many researchers with a mathematical orientation focus on it under the name "competitive analysis". The notation "dynamic vs. static" is substituted here by "online vs. offline".

Comparing an online algorithm to an optimal offline algorithm was first suggested by Sleator et al. (1985) and the term "competive analysis" was coined by Karlin et al. (1988). An online algorithm is called *c-competitive* if the objective function value of the solution produced on any input sequence is **at most** c times that of an optimal offline algorithm on the same input. With the optimal offline algorithm having complete knowledge of the whole input sequence.

"Competitive analysis of online algorithms can be imagined as a game between an online player and a malicious offline adversary. The online player uses an online algorithm to process an input which is generated by the adversary. If the adversary knows the strategy of the online player, he can construct a request sequence which maximizes the ratio between the player's cost and the optimal off-line cost." (cp. Krumke, 2001)

Under very specific assumptions, competitive ratios for the dynamic TSP and the dynamic Traveling Repairman Problem (TRP) are e.g. derived by Jaillet and Wagner (2006). However, for many algorithms directed at solving practical applications such a *worst case* estimation is neither realizable nor very useful. Due to the NP-hardness of all problems, being extensions to the TSP, optimal solutions can usually not be calculated. In addition, the "worst case" scenario may not happen in practice. Information about an algorithm's performance should rather include average performance plus variability measures.

The **second option** can be seen as a simplified modification of the first, without conducting an analytical "worst case" analysis. The performance of some dynamic results is assessed by direct comparison with the optimal static solution. Again, the calculation of such a static optimal solution is usually not possible.

The **third option** for evaluating a dynamic algorithm, which is more practicable than its predecessors, is explained by Mitrovic-Minic et al. (2004). The authors define a "value of information $V(H)$ under heuristic H", which measures the possible gain in solving a dynamic problem ex post heuristically, if all information is known:

$$V(H) = \frac{\dot{x}^H - x^H}{\dot{x}^H}$$

with $\dot{x}^H$ being the best solution of the dynamic instance under heuristic H and x^H being the best solution of the static instance I under heuristic H.

Here, the same heuristic H is used explicitly for calculations on dynamic and static test data, which is probably not the most reasonable decision. According to Psaraftis (1988) and to former explanations, the configuration of dynamic and static algorithms necessitates differences. Here, the calculated value of $V(H)$ will always depend on "how well the corresponding static instance can be solved with a dynamic algorithm which is not supposed to solve such a problem." Therefore, an additional option would be the use of two heuristics: a dynamic one for the dynamic instance and a static one for the corresponding static instance.

The **fourth option** is based on the comparison with another dynamic heuristic that has been proved to produce competitive results for various dynamic data sets. Such an approach is realizable at justifiable expenses and is supposed to yield meaningful results.

The **fifth option** compares the results of a dynamic algorithm with the manually performed planning by human dispatchers. Even if overall manual solution quality can hardly be evaluated, it allows for assessment of a dynamic algorithm's relative solution quality. Therefore, practical usability of this performance measure is also quite high.

2.2 Relevance and Classification of Dynamic Standard Problems

Dynamic transportation problems can be found in several real-life applications where planning decisions have to be made subject to an environment of changing information. This section characterizes the most important real-life planning scenarios and connects these to the underlying theoretical problem definitions.

To classify real-life applications it is reasonable to differentiate between **depot bound** and **depot free** dynamic transportation problems. In the depot bound case, some good has to be transported to or away from a depot. In the depot free case, a depot is not involved. As a second distinctive feature, the application's geographical extension is used. If transportation tasks primarily cover limited operational areas, e.g. "urban areas" with an approximate maximum radius of 50km, we denote the application as **local area**; other applications without geographical limitation are described as **wide area**.

Classification: Depot bound/Depot free and Local Area/Wide Area

A typical representative of the **depot bound** group is the *VRP*, where each vehicle starts and finishes its tour at a prespecified depot, executing a Delivery tour (transportation of goods from the depot to the customers) or a Pickup tour (transportation of goods from the customers to the depot). Since maximum vehicle traveling distance per planning interval is limited, these problems usually occur in **local area** environments. Especially VRPs including Pickup tasks tend to have a dynamic component. This is due to the fact that additional Pickup tasks can be added to a vehicle's tour in the short term just by checking some feasibility constraints, like vehicle capacity or maximum tour duration, once the vehicle is already on its way.

VRP Delivery tours, however, require all Delivery objects to be loaded to the vehicle before the tour starts. If it is decided to add additional Delivery objects to the tour anyway, the vehicle would have to re-visit the depot in order to pick up the additionally needed objects. In practice, Courier services (like UPS or DHL) use the first half of the day to perform static Delivery tasks from the depot to the customers, afterwards (when the vehicle is empty), new (to some extent dynamic) objects are picked up and brought to the depot for further processing. The Vehicle Routing Problem for dynamic Delivery tasks is only conceivable in some special cases, like homogeneous good distribution (e.g. "oil and liquid gases", "beverage cases") and distribution of small products, where a large amount/stock of distributable products can be anticipatorily carried on a vehicle.

The group of **depot free** transportation tasks contains the dynamic *TRP*, where a technician with re-usable repair equipment is sent to dynamically occurring customers. Since just an immaterial "service" is provided to the customers (no transport to or from the depot), new customers can be flexibly included in the repairman's tour. The TRP can be considered as a local area problem.

The depot free group also contains the *Pickup and Delivery Problem (PDP)*, where objects or people have to be immediately transported from a Pickup to a Delivery location, without visiting a depot. We speak of a *Multi Load Pickup and Delivery Problem (MLPDP)* if consolidation of several objects is allowed. Otherwise, if objects have to be transported separately, we denote the problem *Single Load Pickup and Delivery Problem (SLPDP)*. Representative "Pickup and Delivery" applications can be found in local and wide area environments:

In the ***local area*** context, *Taxi or Dial-A-Ride services* deal with dynamic transportation requests of people who wish to be transported from location A to location B. The fast transportation of small parcels, for example between two urban area companies, is summarized in the concept of *Express Mail Delivery Services*. While transportation of passengers is associated with strict capacity constraints, in Express Mail Delivery applications, vehicle capacity can be neglected due to the small size of the parcels. Requests in both applications come along with tight time window constraints. In addition, peoples' transportation usually requires compliance with a maximum ride time duration.

In ***wide area*** environments, especially *occasional transportation* (also referred to as "tramp transportation"), possesses a dynamic component (cp. Section 1.1). Orders, mostly of Single Load type, have to be dynamically assigned to a fleet of moving vehicles, which is spread over the operational area (for example: Europe), producing a transportation schedule with minimal cost. In contrast to local area dynamic problems, the *reaction time* between occurrence of the request and the Pickup time window is usually longer. This is also true for the width of the time windows.

Table 2.1 summarizes the most important dynamic planning problems.

	depot bound	**depot free**
local area	VRP (Pickup Tour for Courier Services)	TRP, MLPDP (Taxi/Dial-A-Ride, Express Mail Delivery)
wide area	—	SLPDP (Occasional long-haul transportation)

Table 2.1: Classification of dynamic real-life problems

Other classifications

Another classification option for dynamic real-life problems was proposed by Gendreau and Potvin (1998). They distinguish between planning problems with "routing" (the need to sequence requests within planned routes) and "no routing". The second criterion is again the problem's geographical extension.

The first category "local area/routing" contains *Courier and Dial-A-Ride services* (see Table 2.2). In addition to our original classification scheme, *emergency services*, like ambulance and police, are also now included (second category: "local area/no routing"). These type of problems can be considered as dynamic transportation tasks, nevertheless their objectives differ. The main challenge is the appropriate re-positioning of a vehicle, once service is completed, in order to reach future requests in a preferably short time. Obviously, "routing" is not an important aspect for this type of problem.

	routing	**no routing**
local area	Courier Services, Dial-A-Ride	Emergency Services
wide area	Less-than-truckload trucking	Truckload trucking

Table 2.2: Classification by Gendreau and Potvin (1998)

More critically, the categorization of Full Truckload wide area transportation into the "no routing" group needs to be questioned. In our real-life problem (cp. Section 5.2), which is derived from a big German freight forwarding company, routing is a **necessary** task. A vehicle's tentative planning schedule usually includes more than one subsequent "routed" request. This differing interpretation may be induced by differences of long-haul transportation tasks in the US and Europe. Due to more agglomeration of economical centers and due to generally smaller distances, the average long-haul travel time in Europe tends to be shorter, allowing more than one transportation task to be scheduled.

Finally, the category "wide area/routing" is filled with the term *Less-than-truckload trucking*, indicating load consolidation for wide area occasional transportation. This is a very rare case to our knowledge. Even very large long-haul trucking companies who perform occasional transportation do not (usually) find requests suitable for consolidation: either the capacity does not allow for consolidation, the time windows are completely different, or suitable requests simply do not occur within geographical proximity. Therefore, as indicated in Section 1.1, less-than-truckload requests that have to be transported in wide area environments are usually fed into the driving routes of an existing – medium term – line operation schedule. This, however, veers away from dynamic short-term planning.

Since many real-life applications fall into the group "local area/routing", the authors perform a sub-classification of this category into the groups "capacitated" and "uncapacitated" as well as "many-to-many" and "one-to-many" (see Table 2.3). A vehicle is "capacitated" if the number of goods that can be loaded onto the vehicle is subject to limitations. From a physical standpoint, each vehicle is somehow "capacitated". However, if the goods are quite small and if there is not the chance of overloading the vehicle during a tour, we denote the associated problem as "uncapacitated". A request is of the type "many-to-many" if two locations (Pickup and Delivery, both different from the depot) are involved. Otherwise, if only a single location (Pickup or Delivery) is involved we speak of "one-to-many".

	many-to-many	**one-to-many**
capacitated	Dial-A-Ride	Feeder systems
uncapacitated	Express Mail Delivery	Courier and Repair Services

Table 2.3: Sub-classification of local-area/routing

For the "many-to-many" case, Gendreau and Potvin (1998) mention *Dial-A-Ride Services* and *Express-Mail-Delivery* with the attributes "capacitated" and "uncapaciated", respectively. The category "one-to-many" includes so-called *Feeder systems* in the "capacitated" case and *Courier and Repair services* in the "uncapaciated" case. A Feeder system is a Dial-A-Ride system with a specific target location, for example an airport or a train station, thus having just one Delivery location. Courier services correspond to the previously mentioned VRP tours; however, in the classification on-hand, there is no accentuation on Pickup tasks.

In the following section, two specific real-life planning scenarios are chosen for further investigation.

2.3 Investigated Problem Settings

This section gives a detailed description of the problems that are investigated in this work: the **dynamic local area (capacitated) MLPDPTW** in Chapter 4 and an extended real-life version of the **dynamic wide area SLPDPTW** in Chapter 5. According to the previous discussion of standard problems, both dynamic problem definitions cover relevant practical applications: Dial-A-Ride services and International Truck Transportation with occasional transportation tasks, respectively.

In the following, a catalog of selected attributes is presented for both problems, especially concerning the underlying *network* structure, the *planning horizon*, attributes of *orders*, *vehicles* and *tours*, as well as the *objective* function (cp. Stumpf, 1998). Tables 2.4 and 2.5 include the attributes of both problem settings.

Major differences occur in the *length of the planning horizon.* The local area planning horizon only covers a 10-hour interval; the planning horizon for the wide area problem, however, has been chosen as 5 weeks. While 10 hours are sufficient to consider a typical working day in the Dial-A-Ride context, a longer horizon is needed in wide area applications, since typical requests come along with a transportation distance that cannot be handled within a single day. In addition, the longer planning horizon allows for the inclusion of time restrictions, like EC social regulations.

Even though both problems cover a Pickup and Delivery problem, there are some differences in the *order characteristic.* The first setting allows for load consolidation, due to "less-than-truckload" order size. In the second scenario, all orders are of type "full truckload", hence there is no option of load consolidation. Nevertheless, some of the second scenario orders may possess several Pickup and/or Delivery locations that have to be processed in a fixed "inner order" sequence.

In the first case, a *homogeneous vehicle fleet* is considered, starting and finishing its tours at a central depot. In the second case, the fleet consists of *heterogeneous vehicles* ("vehicle types"), each with a specific starting position and a specific time of availability. Those vehicles do not have to return to their initial starting position or to any depot at the end of the planning horizon.

Arbitrary order-to-vehicle assignment is only allowed in the first scenario, while in the second scenario, the assignment of an open order requires a vehicle of appropriate type *(restricted order-to-vehicle assignment).*

network	
characteristic	- coordinate network - Euclidean distance
travel times	- constant

planning horizon	
	- rolling horizon - 10 hours + optional overtime

orders	
characteristic	- Pickup and Delivery ("depot free")
location in network	- node
transportation object	- goods
divisibility	- not divisible
time windows	- two-sided time window for Pickup - two-sided time window for Delivery - lower limit (EPT, EDT) = hard constraint - upper limit (LPT, LDT) = soft constraint
sequence of orders	- arbitrary
availability of data	- dynamic order arrival - initial amount of static orders (subject to variation)
size of order	**- less than truckload**
acceptance/rejection	- no rejection, transportation of all orders
geographical extension	**- local area**
transshipment	- no transshipment
number of orders	- 1000
frequency of orders	- singular

vehicles	
number	- 50 (subject to variation) - limited
structure of vehicle fleet	**- homogeneous**
vehicle ownership	- own vehicles
initial location	**- central depot**
applicability	- single day tours - multiple usage
time restrictions	- earliest availability time = hard constraint **- maximum duration** = soft constraint
capacity restrictions	- yes
crew	- one driver mode
driver-to-vehicle assignment	- fixed
order-to-vehicle compatibility	**- no restrictions**

tours	
standard tours	- no
shape of tours	**- start at central depot** **- closed,** vehicles have to return to depot
constraints	- no sequence constraints

objectives	
	minimize: weighted sum of travel time, delay, waiting time, overtime

Table 2.4: Characteristics of the dyn. local area MLPDPTW investigated in Chapter 4

network	
characteristic	- coordinate network - Euclidean distance
travel times	- constant

planning horizon	
	- rolling horizon - 5 weeks

orders	
characteristic	- Pickup and Delivery ("depot free"), with **various Pickup and/or Delivery locations per order and fixed inner order sequence**
location in network	- node
transportation object	- goods, requiring specific vehicle type
divisibility	- not divisible
time windows	- two-sided time window for Pickup - two-sided time window for Delivery - lower limit (EPT, EDT) = hard constraint - upper limit (LPT, LDT) = soft constraint
sequence of orders	- arbitrary
availability of data	- dynamic order arrival - small initial amount of static orders (6.3%)
size of order	**- full truckload**
acceptance/rejection	- no rejection, transportation of all orders
geographical extension	**- wide area**
transshipment	- no transshipment
number of orders	- 14025
frequency of orders	- singular

vehicles	
number	- 953 - limited
structure of vehicle fleet	**- heterogeneous, 5 different vehicle types**
vehicle ownership	- own vehicles
initial location	**- depot free**
applicability	- multi-day tours - multiple usage
time restrictions	**- EC social regulations** = hard constraint **- working time regulations** = hard constraint **- general driving bans** = hard constraint - earliest availability time = hard constraint
crew	**- mixed** (one driver mode, team driver mode)
driver-to-vehicle assignment	- fixed
order compatibility	**- restricted vehicle-to-order assignment**

tours	
standard tours	- no
shape of tours	**- individual vehicle starting position** **- open,** no return to starting position
constraints	- fixed sequence due to order specification - vehicle based time restrictions

objectives	
	minimize: weighted sum of empty travel time, delay, waiting time

Table 2.5: Characteristics of the dyn. wide area SLPDPTW investigated in Chapter 5

The problems also differ in *time restrictions*. The first setting only covers a maximum tour duration (soft constraint), while the second setting covers a set of more sophisticated restrictions, like EC social regulations, working time regulations or general traffic bans (all being treated as hard constraint).

In both cases, the *objective* is to minimize a weighted cost function. However, there are minor differences in the chosen components: in the first case, the cost function includes "total travel time", while in the second case, only "empty travel time" is considered. Due to the maximum tour duration soft constraint, an additional penalty term for "overtime" costs is added in the first scenario.