VALORIZATION

According to the Regulation governing the attainment of doctoral degrees [71], knowledge valorization refers to the “process of creating value from knowledge, by making knowledge suitable and/or available for social (and/or economic) use and by making knowledge suitable for translation into competitive products services, processes and new commercial activities”. So what does this mean? In my opinion two main questions need to be answered: (1) what are the benefits to society of my research, and (2) how easy is it to (directly) apply the results.

In the remainder of this section we will first go deeper into the overarching theme combining the problems discussed in this thesis, followed by a short description of the three problems. After that, the benefits to society and the applicability will be considered.

**Overarching theme** The three problems discussed in this thesis have primarily two things in common: (1) they are based on real-life problems, and (2) they are considered both from a theoretical and practical perspective, with an emphasis on the practical perspective. For the theoretical perspective we consider for instance the complexity of a problem or the worst case performance of an algorithm. For the practical perspective we evaluate the performance of the developed algorithms on realistic instances. With respect to the performance we consider both the quality of the obtained solution and the running time of the algorithm. Note that there is actually no compelling reason why the emphasis is on the practical perspective, except for maybe a personal preference and interest for the practical perspective.

If both the emphasis and the preference is on the practical perspective, why do we still extensively consider the theoretical perspective? One of the reasons is that it can provide valuable information for the practical perspective. For instance, it can give an indication on what kind of algorithm to focus on: if a problem is proven to be NP-hard, it is very unlikely that an exact algorithm exists
that is also efficient. If an efficient algorithm is desired (or maybe even required), it is better to focus on for instance heuristics or approximation algorithms. If an exact algorithm is desired, it will (most likely) have an exponential worst case running time. Also, if we have the worst case performance ratio of an algorithm, we have some kind of indication on the quality of the algorithm; it also provides a mean to compare algorithms with each other.

Note, however, that the theoretical and practical performance of algorithms do not have to coincide: it could be that algorithm A has a better worst case performance guarantee than algorithm B, but performs worse when evaluated on actual instances. A downside of the worst case performance ratio is that it does not tell the whole story, as it is a guarantee that has to hold for all possible instances. It is not uncommon that an algorithm generally performs well, expect for a few (or even a single) artificial instance(s). It would actually be more interesting to have some kind of average case performance ratio. However, it is usually not straightforward to define an average case over all possible instances, and even if it is possible, it is generally hard to analyze. Also evaluating the running time is of importance. It could happen that an algorithm has a good performance with respect to the quality of the solution (this could be both from the theoretical and practical perspective), but if the running time is too high, it will still be of little practical use.

Express Delivery problem The first problem discussed in this thesis is the Express Delivery problem. For this problem we are given a set of packages. Each package is characterized by a start location, end location, volume, and a deadline for delivery. The goal of the problem is to deliver all packages before their deadline, at the minimum cost possible. The transportation of the packages is carried out by trucks. For the cost one can think of for instance the hourly wages of the drivers, gasoline usage, purchase and maintenance cost of the trucks, and so forth. Note that the volume of a single package is generally a lot smaller than the capacity of a truck. Hence, to reduce costs one may want to combine as many packages as possible into a single truck. This however increases the time needed to deliver all the packages in this truck, so that potentially not all deadlines are met. Hence, one main problem is to decide on which packages to consolidate in a truck.
Container Premarshalling problem The second problem discussed in this thesis is the Container Premarshalling problem. This problem occurs at container terminals. One of the main operations at container terminals is the transshipment of containers. Transshipment occurs when a container is moved from one means of transportation (for instance a ship, a train, or a truck) to another means of transportation. In this thesis we solely consider the case where containers are transshipped from one ship to another ship. Note that in this case a container is usually not transshipped immediately, but is first temporarily stored in special area, called the container yard. One of the primary indicators of the efficiency of the operations at a container terminal is the berthing time of a ship, which consists of the time needed to unload and load containers. One way to reduce the loading time is through an operation called premarshalling. The goal of premarshalling is to reshuffle the containers in the container yard into a more desirable lay-out, where containers that need to be loaded first are at the top of their stack, and containers that need to be loaded last are at the bottom of their stack. As the loading plan of a ship usually only becomes available a few hours before arrival, the premarshalling operation should be carried out as fast as possible, which is roughly equivalent to minimizing the number of moves.

Subcontractor Scheduling problem The third problem discussed in this thesis is the Subcontractor Scheduling problem. For the subcontractor Scheduling problem we are given a set of jobs with a dedicated machine for each job, and one additional machine, called the subcontractor, that can be assigned to all jobs. Note that although we use the term machine, one can also think of an actual person. Each job is specified by a duration, release date, and weight. During the time that the subcontractor is assigned to a job, that job is processed twice as fast, resulting in an earlier completion time. If we multiply the amount by which the completion time is reduced by the weight of the job, we obtain the weighted savings of the job. The goal of the Subcontractor Scheduling problem is to find a feasible assignment schedule for the subcontractor that maximizes the weighted savings summed over all jobs. An assignment schedule for the subcontractor is feasible if (1) at each moment in time the subcontractor is assigned to at most one job, and (2) the subcontractor is only assigned to a job between its release date and completion time.
Benefits & Applicability The three problems discussed in this thesis can be categorized as operational planning problems. The main benefits obtained for these problems are of an economical nature. By optimizing the routes for the Express Delivery problem, it might be possible to reduce the number of trucks that are needed and the total distance that needs to be traveled. Optimizing the reshuffling of the container yard for the Container Premarshalling problem has two benefits. Firstly, already by having a better lay-out it might be possible to reduce the loading time, and thus also the berthing time, of a ship. Secondly, by optimizing the reshuffling of the yard, the equipment and available space at the container yard can be used more efficiently. By optimizing the assignment of jobs to the subcontractor for the Subcontractor Scheduling problem, the subcontractor can be put to better use, obtaining a higher return for the company employing the subcontractor.

An additional benefit, especially for the Express Delivery problem, is that the improved solution is more environmental friendly. By reducing the number of trucks, and especially by reducing the number of kilometers that need to be driven, the emission of carbon monoxide is reduced. Note that the economical benefits come as a direct consequence of the objectives that we want to optimize for each problem. For the environmental benefit this is clearly not the case. Hence, the sole objective was to minimize the cost, and obtaining a more environmental friendly solution is just a happy coincidence. However, obtaining more “green” solutions is a topic that is becoming more and more attention and importance.

Now that we have considered some benefits, let us consider the (direct) applicability. The problems considered in this thesis are inspired by real-life problems, and the focus is on the practical side. Unfortunately, this does not mean that the results are directly applicable. While the Container Premarshalling problem is very close to the problem experienced in practice, several abstractions are introduced for the Express Delivery problem and the Subcontractor Scheduling problem. For instance, for the Express Delivery problem each truck is only used on a single connection. This basically means that only the routes for the packages are considered, while the routes for the trucks are ignored. Also, the cost of transshipping a package from one truck to another is not considered. For the Subcontractor Scheduling problem, the processing speed of a job is doubled for
the duration that the subcontractor is assigned to it. Hence, it is assumed that
the subcontractor is equally skilled at each job. Furthermore, it is assumed that
the subcontractor can instantaneously switch between two jobs, i.e., no time is
considered when the subcontractor switches from one job to another. Although
the results cannot be used directly, they still have merit and can be used as a
stepping stone towards solutions tailored to a more specific problem.

If we finally look at the benefits for society, they are not very surprising. Al-
most all research towards operational planning problems focuses on minimizing
the cost, and thus will have the same economical benefits. But one of the require-
ments is to specify the benefits of my research. Therefore, in the remainder of this
section, we consider each of the three problems separately, and discuss some of
the more surprising results.

Results for the Express Delivery problem As already mentioned before, one
important question for the Express Delivery problem is how much consolida-
tion of packages to use. Related to this question is which level of consolidation
possibilities to apply: by restricting the set of routes that packages can use, we
can lower the level of consolidation possibilities, decreasing the difficulty of the
problem. The method we use to reduce the consolidation possibilities is the fol-
lowing: for each problem instance we pick a set of depots, and “promote” them
to hubs. For all packages, we restrict the set of routes that they can use to only
include routes that visit hubs as intermediate depots.

At first glance, restricting the possible routes for a package seems counter in-
tuitive: if we only consider a subset of all possible solutions, we surely cannot
get a better solution, we might even only be able to obtain worse solutions. This
observation would be true, if it would be possible to consider all feasible solu-
tion. However, it takes an impractical amount of time to consider all feasible
solutions, and in our research we limit the available time to solve an instance to
fifteen minutes. Note that even for the most restricted set of routes, fifteen min-
utes is not always enough to obtain the optimal solution. As it turns out, there
are instances for which after fifteen minutes a strictly better solution is found for
a more restricted set of routes than for the set of all possible routes. Note that
this best found solution for the restricted set of routes would also be feasible
for the set of all possible routes, it (or a better solution) has just not been found
within the time limit of fifteen minutes. Hence, in case of a limited amount of time available, it is also advantageous to look at a more restricted set of routes, and thus a more restricted version of a problem.

In a previous section we stated that the theoretical and practical results do not always have to coincide. This is for instance the case if we consider heuristics Independent Shortest Path and Sequential Shortest Path. With respect to the theoretical performance, both heuristics are equally good: if we let $n$ denote the number of packages, we can show that in the worst case both heuristics obtain a solution that is $n$ times as expensive as the optimal solution. If we however consider the practical performance, the cost obtained by Independent Shortest Path is on average roughly twice as high as the cost for Sequential Shortest Path, and in the worst case it is higher by a factor of five. Even with respect to the running time Sequential Shortest Path outperforms Independent Shortest Path.

**Results for the Container Premarshalling problem** While operations at container terminals have been extensively studied, not much research has been done on the premarshalling problem. With respect to a desirable finish lay-out, we consider two variants. For the first variant, called **Priority Stacking**, all lay-outs that do not require any unproductive moves during the loading phase are allowed. For the second variant, called **Configuration Stacking**, only a single pre-specified lay-out is allowed. With respect to the complexity of the problem, it was only known that **Priority Stacking** is NP-hard if the maximum number of containers per stack is at least the total number of containers. As the maximum number of containers per stack is (at most) eight in practice, this did not provide information on the complexity of the problem as it is encountered in practice. We extend the complexity result by showing that both variants are also NP-hard for a fixed stack height of at least six.

With respect to solving the premarshalling problem, the main focus has been on applying heuristics. As far as we know, we are only the second to focus on an exact algorithm, and the first to extensively evaluate its performance. The other exact algorithm is basically evaluated on only two instances. Whereas the other method took 6802 seconds to obtain the optimal solution for one of the instances, our algorithm was able to solve this specific instance within 1 second.
When we started working on the Premarshalling problem, we decided to focus on an exact algorithm that applies Branch-and-Price. However, at that time it was not clear if this approach could actually work. For using this method, several steps need to be performed. The first step is to reformulate the problem, to obtain a so-called Master Problem (MP). This formulation often has many variables, and therefore a problem with only a subset of the variables, called the Restricted Master Problem (RMP), is initially considered. For the RMP the relaxation is solved. As the RMP does not contain all variables, it might “miss” some variables that are used in an optimal solution for RMP. To solve this issue, another problem, called the Pricing Problem (PP), needs to be considered. How the PP looks like depends on the formulation of the MP. By solving the PP it can be determined if a variable is missing from the RMP. If a variable is missing, it is added to the RMP, and the process repeats itself until no missing variables are found. If the found solution to the RMP is integer, we are done. If it is not, branching needs to be applied. Note that the chosen branching rule has an influence on the PP: the previous branching decisions need to be taken into account. If a wrong branching rule is selected, it could happen that the PP becomes difficult to solve when branching decisions are taken into account. Concluding, there are several challenges when applying a Branch-and-Price approach: the MP and the branching rule need to be defined in such a way that the PP can be efficiently solved, also when previous branching decision are taken into account.

**Results for the Subcontractor Scheduling problem** In the case of no release dates (or equivalently, all release dates equal to 0) and no weights (or equivalently, all weights equal to 1), it was known that the Subcontractor Scheduling problem is solved to optimality by always assigning the job with the lowest (remaining) processing time to the subcontractor. We extend this result by showing that this approach remains optimal even if release dates are added to the problem.

For the problem where furthermore weights are added, we do not know an efficient algorithm that determines the optimal solution. Even stronger, we do not even know if the problem is NP-hard. We decided to focus our attention on heuristics. The only difference in the heuristics that we consider is in the selection rule of the next job that is assigned to the subcontractor. We consider
four selection rules, namely selecting the job with: (1) the lowest (remaining) processing time (called Proctime), (2) the highest weight (Weight), (3) the highest ratio of weight to (remaining) processing time, also known as the Smith ratio (Smith), and (4) the highest value for (remaining) processing time multiplied by the weight (Profit). Proctime is added because it is optimal in the case of no weights, and Weight is considered because analogously to Proctime it only considers one input dimension. Smith determines the optimal solution for a related problem, minimizing the weighted sum of completion times on a single machine.

Our belief was that Smith, or maybe Profit, would have the best performance guarantee, as they consider both the (remaining) processing time and weight to determine the next job. However, as it turns out Weight has the best performance guarantee, obtaining savings of at least two thirds of the optimal savings. The guarantee is (at most) one half for Smith and $\frac{1}{n}$ for Profit, where $n$ is the number of jobs. For Proctime the guarantee is $\frac{1}{2}n^{-1}$, meaning that for 11 jobs the guaranteed savings is less than 0.1% of the optimal savings.

As noted before, the performance guarantee must hold for all instances. Therefore the practical results might give a different view. Besides the four rules already mentioned, we consider five more selection rules: picking the next job at random, and giving higher importance to the weight for Smith and Profit, by either taking the weight squared or weight cubed. As it turns out, the practical results are in line with the theoretical results. Overall Weight performs best, followed by the variants of Smith and Profit where the weight cubed is considered. What is somewhat surprising is the performance of Proctime. It performs only slightly better than picking the next job at random, but a lot less than the next worst algorithm.