Thesis for the Degree of Doctor of Philosophy

Mathematical Multi-Objective Optimization of the Tactical Allocation of Machining Resources in Functional Workshops

Sunney Fotedar

Division of Applied Mathematics and Mathematical Statistics
Department of Mathematical Sciences
Chalmers University of Technology
Göteborg, Sweden 2023

Mathematical Multi-Objective Optimization of the Tactical Allocation of Machining Resources in Functional Workshops
Sunney Fotedar
Göteborg 2023
Email sunney@chalmers.se
ISBN 978-91-7905-902-6

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 5368
ISSN 0346-718X

The research is financially supported by VINNOVA, Chalmers University of Technology, and GKN Aerospace Sweden AB.

Division of Applied Mathematics and Mathematical Statistics
Department of Mathematical Sciences
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone +46 (0)31 772 4275

Online version: https://research.chalmers.se/person/sunney

Typeset with LaTeX
Printed by Chalmers digitaltryck
Göteborg, Sweden 2023

Mathematical Multi-Objective Optimization of the Tactical Allocation of Machining Resources in Functional Workshops

Sunney Fotedar

Division of Applied Mathematics and Mathematical Statistics
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

In the aerospace industry, efficient management of machining capacity is crucial to meet the required service levels to customers and to maintain control of the tied-up working capital. We introduce new *multi-item*, *multi-level* capacitated resource allocation models with a medium–to–long–term planning horizon. The model refers to functional workshops where costly and/or time- and resource-demanding preparations (or qualifications) are required each time a product needs to be (re)allocated to a machining resource. Our goal is to identify possible product routings through the factory which minimize the maximum excess resource loading above a given loading threshold while incurring as low qualification costs as possible and minimizing the inventory.

In *Paper I*, we propose a new bi-objective mixed-integer (linear) optimization model for the *Tactical Resource Allocation Problem* (TRAP). We highlight some of the mathematical properties of the TRAP which are utilized to enhance the solution process. In *Paper II*, we address the uncertainty in the coefficients of one of the objective functions considered in the bi-objective TRAP. We propose a new bi-objective robust efficiency concept and highlight its benefits over existing robust efficiency concepts. In *Paper III*, we extend the TRAP with an inventory of semi-finished as well as finished parts, resulting in a tri-objective mixed-integer (linear) programming model. We create a criterion space partitioning approach that enables solving sub-problems simultaneously. In *Paper IV*, using our knowledge from our previous work we embarked upon a task to generalize our findings to develop an approach for any discrete tri-objective optimization problem. The focus is on identifying a representative set of non-dominated points with a pre-defined desired *coverage gap*.

Keywords: Capacity planning, Discrete bi-objective and tri-objective optimization, Robust efficient solutions, Decision-making, Representation of the set of non-dominated points, Coverage gap.

List of publications (or manuscripts)

This thesis is based on the work represented by the following four papers:

I. **Fotedar, S.**, Strömberg, A.-B., Almgren, T. (2022). Bi-objective optimization of the tactical allocation of job types to machines. Mathematical modeling, theoretical analysis, and numerical tests. *International Transactions in Operational Research*, 30(6), pp. 3479–3507 Fotedar et al. [2023a]

II. **Fotedar, S.**, Strömberg, A.-B., Åblad, E., Almgren, T. (2022). Robust optimization of a bi-objective tactical resource allocation problem with uncertain qualification costs. *Autonomous Agents and Multi-Agent Systems*, 36, pp. 1–31. Fotedar et al. [2022]

III. **Fotedar, S.**, Strömberg, A.-B., Almgren, T., Cedergren, S. (2023). A criterion space decomposition approach to generalized tri-objective tactical resource allocation. *Computational Management Science*, 20(1), pp. 1–28. Fotedar et al. [2023b]

IV. **Fotedar, S.** and Strömberg, A.-B (under review at European Journal of Operational Research). A method to identify a representation of the set of non-dominated points for discrete tri-objective optimization problems

Conference papers not included in this thesis or for evaluation:

V. **Fotedar, S.**, Almgren, T., Cedergren, S., Strömberg, A.-B., Patriksson, M. (2019). A Mathematical Optimization of the Tactical Resource Allocation of Machining Resources for an Efficient Capacity Utilization in Aerospace Component Manufacturing. *Proceedings of the 10th Aerospace Technology Congress, October 8–9, 2019, Stockholm, Sweden*, doi: ggcf5t.

VI. **Fotedar, S.**, Almgren, T., Wikner, J., Strömberg, A.-B., Cedergren, S. (2020). A decision-making tool to identify routings for an efficient utilization of machining resources: the decision makers' perspective. *PLANs Forsknings- och Tillämpingskonferens 2020, Sweden*.

Author contributions

All the papers are written and conceptualized by me. The co-authors helped in improving manuscript quality and of course, trained me to develop scientific rigor.

I. I developed the tactical resource allocation model, and the modified *bi-directional* ϵ-constraint method used to solve the problem. I implemented the code, ran the experiments, and wrote the manuscript.

II. I suggested the new extension to a robust formulation for the given bi-objective MILP. I suggested a new *robust efficiency* concept for our model and a new 3-*stage approach* to find relevant robust efficient solutions.

III. I developed the tri-objective mixed integer optimization model which includes inventory. Furthermore, I suggested splitting the projected two-dimensional criterion space and ran tests on relevant industrial instances. Development of code and tests are done by me.

IV. I proposed the algorithm to identify a representation of the set of non-dominated points. I developed the code and tested it on relevant instances.

Abbreviations

- **APICS**: currently known as the Association for Supply Chain Management
- **AWT**: Augmented Weighted Tchebycheff
- **BB**: Balanced Box
- **BOIP**: Bi-Objective Integer Programming
- **BOMIP**: Bi-Objective Mixed Integer Programming
- **BOMILP**: Bi-Objective Mixed Integer Linear Programming
- **DM**: Decision-maker
- **FRE**: Flimsily Robust Efficient
- **HPC**: High Pressure Compressor
- **HRE**: Highly Robust Efficient
- **ILP**: Integer Linear Programming
- **LP**: Linear Programming
- **LPC**: Low Pressure Compressor
- **LPT**: Low Pressure Turbine
- **MILP**: Mixed Integer Linear Programming
- **MIP**: Mixed Integer Programming
- **TRAP**: Tactical Resource Allocation Problem
- **MPC**: Manufacturing, Planning and Control
- **MRP**: Material Requirements Planning
- **MOOP**: Multi-Objective Optimization Problem
- **NDP**: Non-dominated point
- **PRO**: Pareto Robust Optimal
- **RE**: Robust Efficient
- **SO-RO**: Single-Objective Robust Optimization
- **QSM**: Quadrant Shrinking Method
- **TOIP**: Tri-Objective Integer Programming

Acknowledgements

I would like to thank my supervisor Prof. Ann-Brith Strömberg for her constructive criticisms and guidance throughout this period. Most importantly, acting as a good role model in mathematical writing. I would also like to thank my industrial supervisors Dr. Torgny Almgren and Dr. Stefan Cedergren for their continued assistance/guidance in defining the problem for GKN Aerospace, and sharing valuable knowledge in logistics, and machining aspects of the problem. I would also like to thank Prof. Joakim Wikner (Jönköping University), and Prof. Michael Patriksson (Chalmers) for their contributions. Special thanks to fellow Ph.D.-students in the optimization group. I am also grateful to my previous teacher who inspired me to take up optimization as a research topic: Prof. Dag Haugland (University of Bergen, Norway) and Prof. Stein Wallace (NHH Norwegian School of Economics) for his discussions on modeling under uncertainty. I am also grateful for the support from Prof. Annika Lang, and other staff at the Mathematical Sciences department at Chalmers/GU. I appreciate all the help from GKN Aerospace (especially, Håkan Colliander and Dr. Karin Thörnblad, and other members of the logistics department).

Finally, I would like to extend my heartfelt gratitude to my wife, Dr. Indu Dhar, and my son Neel, for their unwavering love and support. They have been a constant source of inspiration and have generously given me the freedom to pursue my own career path in life.

Sunney Fotedar, Göteborg, 2023

Contents

1 Introduction

The field of **M**anufacturing, **P**lanning, and **C**ontrol (MPC) as defined in APICS dictionary [Blackstone Jr., 2013, p. 99] is described as *a closed-loop information system which includes planning functions of production planning, sales and operations (S&OP) [Blackstone Jr., 2013, p. 154], master production scheduling [Blackstone Jr., 2013, p. 101], material requirements planning [Blackstone Jr., 2013, p. 103], and capacity requirements planning*. Due to the increased complexity of businesses and production methods, most of medium- and large-sized companies have implemented computerized planning systems in the past few decades. Generally, these are transactional systems helping to track flows of material. Therefore, it maintains updated procurement and manufacturing information on each planning decision. However, unless these tools are combined with *mathematical optimization*, the chances of getting the best possible solution are minimal (and not guaranteed at all).

Mathematical optimization is a topic/subject in applied mathematics that deals with finding the best possible solution to a *decision problem* (although, sometimes only dealing with the feasibility problem is sufficient). The definition of *best* can vary depending on the definition of the *objective function(s)* of the optimization problem. In 1960s and 1970s, mixed integer programming (MIP) models became popular among operations research practitioners who tried to tackle simple planning problems using MIPs. However, as the size of the problem instance grows solving a MIP to optimality becomes computationally demanding. Many state-of-the-art commercial solvers have made solving MIPs relatively easier as compared to a few decades ago. The adoption of mathematical optimization has increased for industrial process planning. In this work, a few novel mathematical optimization models are proposed for allocating machining resources to jobs for medium-to-long-range planning horizons for GKN Aerospace Engine Systems (GKN for short) in Trollhättan, Sweden. The model is intended to assist engineers and planners to make decisions regarding product routings in the factory.

1.1 Case Company: GKN Aerospace Engine Systems

GKN is a leading supplier of aircraft engine parts. GKN's products are present in almost all major commercial aircraft. The products manufactured at the Trollhättan factory in Sweden include fans at the front of the jet engines or gas turbines, rotors, stators, and other turbine structures. Rotation and a high-temperature difference between different parts of the engine put high, in many cases extreme, quality demands (tight tolerance limits). The capital-intensive production at a large aerospace tier-1 supplier like GKN is generally influenced by expensive materials, long supply lead times, a large product mix (see Lewestam and Mäki [2015]).

1.2 Production context

Manufacturing is performed in multiple steps, such as cutting (milling, turning, drilling, and grinding), welding, assembly, heat/surface treatments, and control/measurements. For cutting, GKN has a variety of production resources (machines) with different functions. The factory is organized in several functionally oriented production shops (so-called *functional workshops*) [Blackstone Jr., 2013, p. 70][1], and most of the production resources are shared by several products. Each production shop is organized as a *job-shop* [Blackstone Jr., 2013, p. 87], where similar types of machines are placed in proximity to each other. A complication is that it is, in practice, impossible to physically move machines, as they are bulky and fixed into the ground in 2–5 meters deep pits to avoid mechanical vibrations. Thus, the factory as such can only to a very limited extent be adapted to changes in the product mix. It is therefore not possible to maintain perfect flows of parts through the factory over time. Hence, managing capacity, especially machining capacity (since it takes up a large share of the total lead times) is crucial for GKN.

1.3 Capacity management

The focus of our research is on *capacity management* [Blackstone Jr., 2013, p. 22] which is defined as *the function of establishing, measuring, monitoring, and ad-*

[1]A factory configuration in which similar operations are grouped together. An alternate term is a job-shop layout

justing limits or levels of capacity in order to execute all manufacturing activities. Capacity management is also referred to as a *response to variation in demand*. However, demand is generally not the only source of variation; another source that is internally generated is due to additional capacity requirements caused by *re-works*, which is the reprocessing done to salvage a defective item or part (see [Blackstone Jr., 2013, p. 152]). Our interest in aspects of capacity management is because it helps the planners to absorb some of these variations.

At GKN, a hierarchical approach is used to plan for the machining capacity, instead of using one big monolithic model. In a hierarchical approach, the decisions made at the top levels influence (set the boundary conditions/constraints) the decisions made at lower levels. Within this hierarchical approach, a feedback loop can help to continuously improve the efficiency of the planning system by appropriately adjusting the control parameters.

Figure 1.1 illustrates the decomposition of capacity management into the three levels, *capacity strategy*, *capacity planning*, and *capacity control*. The capacity strategy deals with the decision regarding investment in new machines and identifying/modifying product structures. This is done between 2–6 years in advance. Capacity strategy requires input from manufacturing experts to establish the bill of material (see [Blackstone Jr., 2013, p. 15]), which is simply a list of parts, sub-assemblies and raw materials required to form a final product, and the *operations list*, that details the method of manufacture of a part and its sequences.

The output from the capacity strategy level defines the solution space for *capacity planning*, also known as *rough-cut capacity planning* (see [Blackstone Jr., 2013, p. 153]). Capacity planning deals with tactical allocation decisions made 1–4 years in advance. This identifies product routings which include the operations performed, their sequences, and machines involved to process them (see [Blackstone Jr., 2013, p. 153]). It is also necessary for a functional workshop to prepare/qualify more than one possible routing for each product, rendering flexibility to the production planners.

The next level is the *capacity control* [Blackstone Jr., 2013, p. 22], which is the process of measuring output from production and comparing it with the actual capacity plan. There is usually a difference between the two, and necessary corrective measures are required to prevent serious delivery issues. The input is all possible qualified (approved) routings. These alternative routings [Blackstone Jr., 2013, p. 6] provide the necessary flexibility to the planners to tackle any short-term demand variability. Certain performance indices are tracked at each level, and a feedback loop that goes up one level could be used to adjust different control parameters. This gradually improves the accuracy of mod-

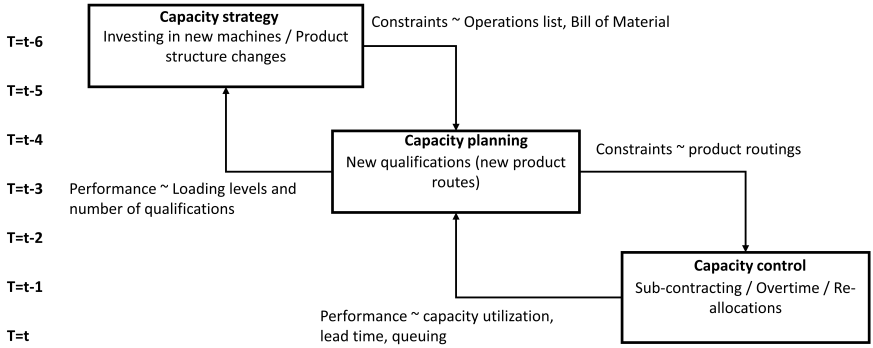


Figure 1.1: Hierarchial planning framework with different capacity sources and respective feedback loops (time discretization in years)

els for capacity planning as some of the control parameters are appropriately adjusted to produce desirable changes.

1.4 The need for a tactical resource allocation model

The decision regarding where to process products/parts is generally made at the time of the introduction of new products to a factory (or less frequently, with a significant change in the production capacity). The process of introducing a new product is inevitably linked with resource allocation decisions in a functional workshop. These tactical resource allocations referred to in this text should not be mistaken for the short-term resource allocations done when choosing between resources (among several qualified resources) while scheduling. The latter is commonly addressed in the industry (see an example from GKN, Thörnblad et al. [2015]). At the time of introduction of new products, manufacturing experts decide the operations list for a product and where respective operations will be performed. This process involves qualifying machines for a new job, running simulations and physical tests to check the quality of features (for example, how accurate (e.g. roundness) was a certain hole drilled in a job) produced. Thus, once a product's operations are assigned to certain machines, changing it is a very costly, and time-consuming, activity.

A decision-making tool that supports GKN in resource allocations for medium-to-long term planning horizon is needed. Our proposed model provides the *routings* to be used by products, and suggestions of new qualifications to be performed either for new or old products. A general framework for the

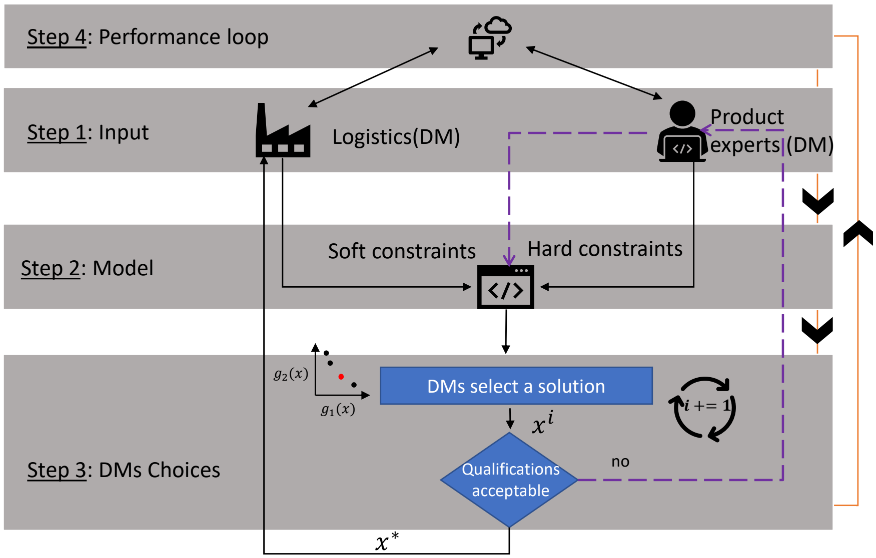


Figure 1.2: Framework for resource allocation decision-making tool

decision-making tool is presented in Figure 1.2. Although the focus in this thesis is mainly on the model (i.e. step 2), it is still important to understand how the results are going to be utilized. All four steps are part of a continuous procedure. In the first step, the input regarding constraints is provided by logisticians and product experts, who are the two types of decision-makers (DMs) involved. *Soft constraints* are related to decision makers' preferences, and can be modelled either as objective functions or as hard constraints accompanied with lexicographic minimization (see Definition 5 *routing efficient solution* in Paper I). *Hard constraints* are related to the feasibility of allocating a job to a machine, and to capacity limitations of machines. *Hard constraints* are modeled as constraints in the optimization model. In the second step, a multi-objective optimization model is solved. Then, an efficient solution is chosen and further analyzed by the product experts to check if the new qualifications are indeed feasible, using simulations or lab experimentation. If the qualification is not acceptable, then product experts add new constraints to the model, and a new (or slightly different) set of efficient solutions is identified. The final solution is sent to the logistics department responsible for utilizing these new routings in its scheduling software. The fourth step is about tracking the effect of new qualifications on lead times, capacity utilization, and other performance criteria.

1.4.1 Routings

The term *routings* is sometimes used as an alternative term for *instruction sheet* and *bill of operations*, which detail the method of manufacture of a particular product. Routings includes a list of operations to be performed, along with details about the machines in which the operations must be performed. For example in Figure 1.3, a product's/part's routings are illustrated. It begins with raw materials released from the inventory. Then, all such types of raw materials are sent to machine M_1 where the first operation is performed for all the three routings (R_1, R_2, R_3). The second operation can be performed in three different machines $M_2(\text{in } R_1), M_3(\text{in } R_2)$ and $M_6(\text{in } R_3)$. Afterwards, the raw materials are sent to M_3 for the third operation, and to machine M_2 for the fourth (the last operation). All the finished products/parts are sent to the final inventory of finished goods. The three different routings differ only in one operation, i.e. the second operation. The two dashed rectangles enclosing several machines represent two different shop floors (physical locations in the factory). Note that the same product/part may visit the same machine multiple times in the process of getting transformed into the finished product (for example, in Figure 1.3, machine M_2 is visited twice in routing R_1). The three routings considered are shown at the bottom of Figure 1.3. It is generally well-known that having several alternative routings for a product provides necessary *capacity cushion* for managing short-term demand variations, especially, when machines are shared among products. Thus, it is beneficial to have several routings qualified for a product.

Each operation has to be qualified for a machine, which requires a significant one-time cost in the form of man-hours for programming the control systems and of buying new fixtures or tools. These new qualifications also require approval from the customers. Thus, it requires time as well as money to prepare new routings. Hence it should be done well-in-advance, and with some thought. GKN has 120 machines and thousands of different parts with at least 5–10 operations, hence, the number of feasible allocations/routings are simply too many to enumerate and a mathematical analysis of the problem is therefore necessary.

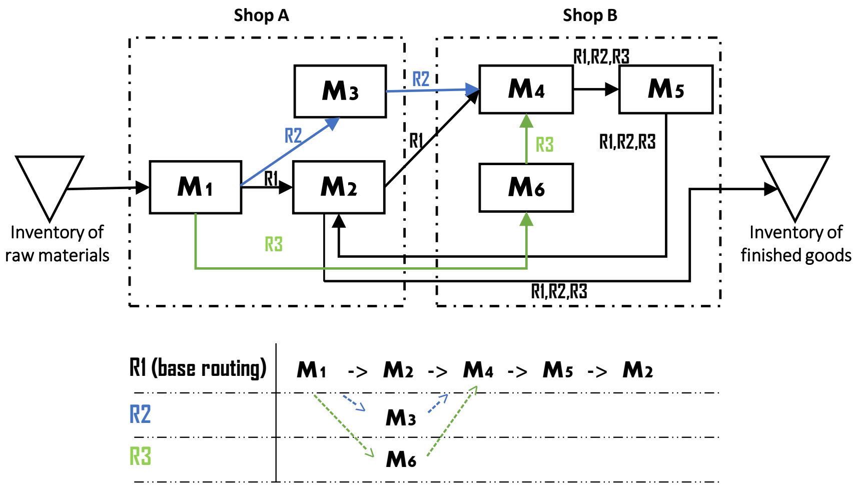


Figure 1.3: Routings for a product. The base routing is the routing that is used most frequently

1.5 Previous work

The research field of production planning is broad. We provide a brief overview of the field before diving into the specific variant of the production planning problem studied and solved in this work.

One popular way of classifying production planning models is by acknowledging the considered time-horizons. This simplifies to some extent the decision variables and parameters used in the model. Several authors (e.g. Min and Zhou [2002], Gupta and Maranas [1999]) classify production planning problems as *strategic*, *tactical*, or *operational*. Our focus is on tactical models, since we make medium-to-long term capacity planning decisions. To the best of our knowledge the most recent review of tactical level mathematical production planning models is done by Díaz-Madroñero et al. [2014]. The authors have identified the following categorizations:

(a) *Number of products and number of levels of the product structure*

The number of products/parts being manufactured; their levels refer to whether a product has a flat bill of material (BOM) or BOM with multiple levels consisting of various sub-assemblies. Different variants of multi-level products/parts (e.g. series, assembly, general, arborescence) are described in [Pochet and Wolsey, 2006, Ch. 13]

(b) *Time periods*

Deals with the size of the time-buckets used. In a *small* time-bucket, the time period is long enough to produce only one part/item, whereas in long time-buckets, either multiple items can be produced or a final product consisting of multiple items. In Transchel et al. [2011], the authors have considered both types.

(c) *Nature of the demand*

The demand uncertainty is mainly tackled by stochastic approaches, and less commonly by robust or fuzzy approaches. Some of the research done in stochastic demand production planning models are Genin et al. [2008] (added noise to demand patterns), Wei et al. [2011] (uses robust approaches with interval uncertainty) and Chen and Huang [2010] (uses fuzzy approaches).

(d) *Capacity constraint classes*

Numerous combinations of capacity constraint classes are comprehensively reviewed in [Díaz-Madroñero et al., 2014, p. 5176, Table 6]. The classes include inventory, supply, production resources, and transport services.

(e) *Types of objective functions*

The most common type of objective function minimizes costs/time (processing time, set-up time, and fixture costs) (see Bradley and Glynn [2002], Van Mieghem [2003]). However, using such an objective function has drawbacks since most of the cost measures rely heavily on the used accounting principles, which are sometimes misleading as highlighted in Myrelid and Olhager [2019]. Some of the other objectives considered in the literature are minimizing backlogs, maximizing throughput, and maximizing utilization.

Apart from the above-mentioned categorizations, there are numerous other extensions. In *unrelated* parallel machine problems Garey and Johnson [1979], the processing times of jobs (tasks/operations) at machines are not related to each other and depend on the machine in which it is being processed. On the contrary, in *related* parallel machine problems, the processing times of jobs are independent of the machine in which they are being processed.

1.6 Scope

The aim of our project is to create a *multi-item*, *multi-level*, *big time-bucket*, *capacitated*, *unrelated parallel machine* tactical resource allocation model. The focus of our models is resource loading, i.e. planning the allocation of capacity of machines for a time frame where reliable weekly or daily demand predictions do not exist. Consequently, short-time buckets are not relevant to our models. Instead, time discretization employs long time steps (a quarter of a year) wherein it is reasonable to assume a constant material flow. For example, a product or a part $P1$ requires a set of operations, for instance, $\{\mathrm{OP100}, \mathrm{OP200}, \mathrm{OP300}\}$. A *job-type* is a combination of a part type and an operation. The orders of job types, that is, each element of the set of 2-tuples $\{(P1, \mathrm{OP100}), (P1, \mathrm{OP200}), (P1, \mathrm{OP300})\}$, must be allocated to machines in each time period over a long time horizon (1–4 years) with quarterly time-buckets.

2 Problem description and modelling

In this chapter, the so-called *Tactical resource allocation problem* (TRAP) is discussed. We present the basic model that is presented in Paper I and further developed in the remaining papers.

In Table 2.1 routings for a dummy production system are illustrated with three machines ($k = 1, 2, 3$) and two operations milling and turning are performed on a single product. In the first time period $t = 1$, milling is done in machines $k = 1$ and $k = 3$, in the second time period the same operation is done in machines 1 and 2, and in the third time period it is done only in machine 2. For machine 2, the box around the M indicates that the milling operation is to be qualified and performed in time-period 2. Similarly, for machine 3 the turning operation is to be qualified in time-period 2 and performed in time-period 3. This qualification requires a one-time cost which includes the cost of new fixtures and the cost of time spent on programming the control systems. The qualification must be done either before or at the beginning of the time period when it is to be used.

	$k = 1$		$k = 2$		$k = 3$	
$t = 1$	M	T		T	M	
$t = 2$	M	T	$\boxed{\text{M}}$	T		$\boxed{\phantom{\text{M}}}$
$t = 3$			M	T		T

Table 2.1: Routings for a single part/product: M (milling) and T (turning) indicate time periods (t) when machines (k) are used for the respective purpose; $\boxed{\phantom{\text{M}}}$ indicates time period and machine qualification for milling and turning, respectively.

2.1 The feasible set: a non-mathematical description

Constraints in an optimization model limit the domain of feasible solutions (decision variables) acceptable to the planner. However, in [Wierzbicki et al., 2000, Chapter 5], it is argued that in the real-world many constraints are divided into so-called *soft constraints*, and *hard constraints*. The authors suggest modeling soft constraints as additional objectives for the optimization problem, and hard constraints as constraints of the mathematical model. In this section, a non-mathematical description of the hard constraints is presented

(a) *Demand*

Each time bucket is a quarter of a year, which is significantly larger than the total lead time of products. Hence, the demand in each time period must be satisfied within the same time period.

(b) *Routing limitations* (τ)

These types of constraints ensure that it is not allowed to allocate orders of the same job-type to more than a user-defined number of machines in each time-period, denoted by $\tau \geq 1$. These constraints keep the product flow less complex for the production planners. For instance, in Figure 2.1, we assume a part type $P1$ (represented by a red-node) requiring two operations $\{\mathrm{OP100}, \mathrm{OP200}\}$, and the two corresponding job-types are $(P1, \mathrm{OP100})$ and $(P1, \mathrm{OP200})$ (see the black rectangles at the top of Figure 2.1). If $\tau = 2$, then only two machines are allowed to perform job-types $(P1, \mathrm{OP100})$ and $(P1, \mathrm{OP200})$ during the same time-period. Hence, there are at most four different feasible routings for product $P1$ (in Figure 2.1 these routings are $R2, R3, R4, R5$ marked with black-arrows). However, if $\tau = 3$, the number of possible routings can be nine (three machines for each job-type). Increasing the value of τ results in a greater chance to balance the resource loading; however, having too many routings may result in a complicated product flow that is not suitable to the planners. In Figure 2.1, it is evident that if the value of τ is increased to three, the routings $R1$ and $R6$ (blue-dashed arrows) are allowed as well. The end-user should provide a limitation on the parameter τ.

(c) *Qualification costs ($\boldsymbol{\beta}$) and related limitations (γ)*

These types of constraints ensure that a given job-type must be qualified for a machine before planners can start using them. The qualification

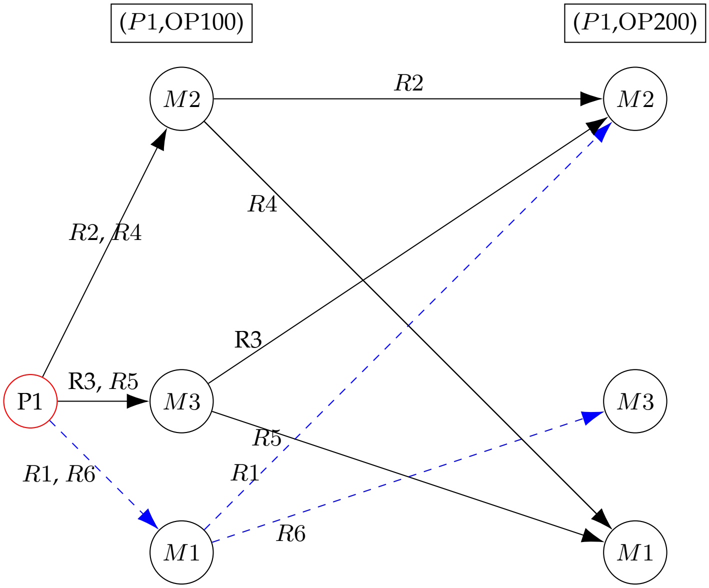


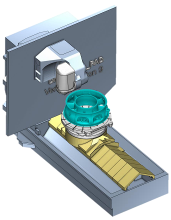
Figure 2.1: Routing limitations: $P1$: a product/part node, $\{M1, M2, M3\}$ is the set of machines.

cost associated with qualifying jobs for machines may be in the form of time spent by manufacturing experts to program the control systems, or buying new fixtures or tools. The exact costs for qualifying a machine for a job type are not known a priori, and an accurate prediction requires detailed simulation work by the engineering team. Hence, ordinal numbers (or levels) will be used.

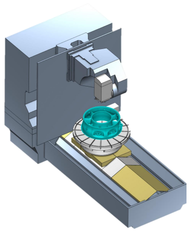
In Figures 2.2a and 2.2b the two multi-task machines capable of performing both milling and turning operations are illustrated. One of the significant differences between the two is that the one to the left (Figure 2.2a) has smaller diameter turning table as compared to the one to the right (Figure 2.2b). Hence, given that all the other operational conditions are the same, it should be technically possible to move some jobs from the machine to the left to the machine to the right. However, there will still be a not too high cost associated with it. Note that apart from diameter of the parts/products, and the turning tables, there are many other part/product features and machine capabilities that have to match for the allocation to be feasible. In this work, it is assumed that such information is available. The multi-task machines illustrated in Figure 2.2c and Figure 2.2d have the so-called *B-axis*, that is, they are inclined at $45°$. This is the only difference between the machines in Figure 2.2a and

Figure 2.2c, otherwise they are the same size, both having diameter of the turning table 1.25 m. The turning table in the machine in Figure 2.2d is larger than that of Figure 2.2a–Figure 2.2c. There are both benefits and drawbacks of moving a job from the multi-task machine in Figure 2.2a or Figure 2.2b to any of the B-axis machines in Figure 2.2c or Figure 2.2d. A benefit is that the productivity is increased due to the use of lower cutting parameters, such as *feed rate* or *depth of cut*, to reach the same level of quality; hence, the operations are faster. Furthermore, inclined machines are also more robust w.r.t. the production of more accurate features on products/parts; hence, less chances of need for re-works. So, both productivity and robustness are increased in the machine with the $45°$ inclined axis. However, the downside is that the inclined spindle head may reduce the accessibility to certain sections of the part/product, thus, making it incapable of producing certain types of product features. In Figure 2.2e and Figure 2.2f, the two vertical lathes are capable of performing only turning operations. All of this information has to be encoded appropriately to be used as parameters in the mathematical model.

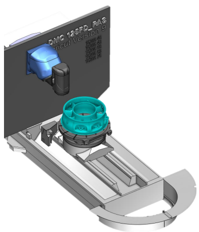
There is also a limitation on the total number of new qualifications to perform in each time-period. This is a result of the limited number of trained technical experts. This limitation is denoted by $\gamma \in \mathbb{Z}_+$.



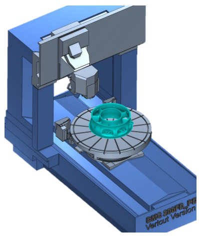
(a) A multi-task machine performing both turning and milling operations with a **small-sized turning table**



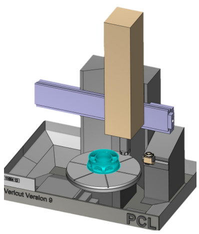
(b) A multi-task machine performing both turning and milling operations with a **medium-sized turning table**



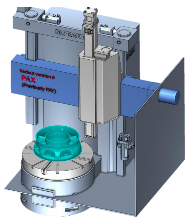
(c) A multi-task machine with inclined spindle head performing both turning and milling operations with a **small-sized turning table**



(d) A multi-task machine with inclined spindle head performing both turning and milling operations with a **large-sized turning table**



(e) A vertical lathe machine performing turning operations with a **large-sized turning table**



(f) A vertical lathe machine performing turning operation with a **medium-sized turning table**

Figure 2.2: Different machine alternatives

2.2 Problem definition: a mathematical description

The deterministic version of the tactical resource allocation problem (TRAP) addressed in Paper I is defined in this section. It is called deterministic, as the values of all the parameters are (assumed to be) known. The notation used for the model is described in Table 2.2.

Definition 2.1 (Tactical Resource Allocation Problem (TRAP)). *Given a set $\mathcal{J}$ of job-types (tasks) and a set $\mathcal{K}$ of machines, let p_{jk} be the average processing time (including set-up time) of job-type $j \in \mathcal{J}$ when performed in a compatible machine $k \in \mathcal{K}_j \subseteq \mathcal{K}$. Each machine $k \in \mathcal{K}$ has the capacity C_{kt} (time units) in time-period $t \in \mathcal{T}$ and a relative loading threshold $\zeta_k \in [0, 1]$. The demand a_{jt} of each job-type $j \in \mathcal{J}$ in time-period $t \in \mathcal{T}$ must be met. The number of machines allocated to the same job-type in each time-period must not exceed the value of the parameter $\tau \in \mathbb{Z}_+$. For assignments (j, k), such that $k \in \mathcal{N}_j$ and $j \in \mathcal{J}$, so-called qualifications are required, which generate additional one-time costs. It holds that $\mathcal{N}_j \subseteq \mathcal{K}_j$ for all $j \in \mathcal{J}$; for the case of a new job-type (associated with a new product) j, $\mathcal{K}_j = \mathcal{N}_j$ holds. For a job-type $j \in \mathcal{J}$, the machines in the set $\mathcal{K}_j \setminus \mathcal{N}_j$ do not require any qualifications. The total number of qualifications performed per time-period t may not exceed the value of the parameter $\gamma \in \mathbb{Z}_+$. The objectives considered are to minimize the sum (over time-periods) of maximum excess resource loading above a given threshold ζ_k over each machine $k \in \mathcal{K}$ and to minimize the sum of qualification costs incurred.* □

Excess resource loading (g_1) The objective function is defined by g_1, to be minimized, considers the sum over the time-periods $t \in \mathcal{T}$ of the *excess resource loading of the machines* (i.e. $n_t \geq 0$), which is defined as the maximum (over the machines) ratio between the allocated machining hours and the available hours (i.e. $\frac{1}{C_{kt}} \sum_{j \in \mathcal{J}} p_{jk} x_{jkt}$) minus the loading threshold $\zeta_k \in [0, 1]$ for the machine. The thresholds ζ_k are provided by the users. Therefore, in a solution $\mathbf{y}$ that minimizes the objective g_1, the equality $n_t = \max\left\{0; \max_{k \in \mathcal{K}} \left\{\frac{1}{C_{kt}} \sum_{j \in \mathcal{J}} p_{jk} x_{jkt} - \zeta_k\right\}\right\}$ will hold for $t \in \mathcal{T}$. In the context of a *bi-objective mixed integer programming* (BOMIP) problem, it is defined by (2.1a), (2.2d), (2.2g), and (2.2j), below.

The practical motivation for employing this objective function is to avoid—for each machine in each time-period—that the planned loading level (i.e. $\sum_{j \in \mathcal{J}} p_{jk} x_{jkt}$) exceeds the user-defined threshold ($C_{kt}\zeta_k$). As a result, this will help in maintaining some capacity buffers to be used when there is a short-term demand variation, which in turn implies that the queuing times are kept at a minimum.

Table 2.2: Notation for the tactical resource allocation problem (TRAP)

Sets	Description
$\mathcal{J} = \{1, \ldots, J\}$	set of job-types to be performed on the products
$\mathcal{K} = \{1, \ldots, K\}$	set of machines
$\mathcal{K}_j \subseteq \mathcal{K}$	set of machines feasible for job-type $j \in \mathcal{J}$
$\mathcal{N}_j \subseteq \mathcal{K}_j$	set of machines feasible, but not qualified for job-type $j \in \mathcal{J}$
$\mathcal{T} = \{1, \ldots, T\}$	set of time-periods
Variables	**Description**
$x_{jkt} \in \mathbb{Z}_+$	number of orders of job-type $j \in \mathcal{J}$ performed in machine $k \in \mathcal{K}_j$ in time-period $t \in \mathcal{T}$
$s_{jkt} \in \{0, 1\}$	equals 1 if job-type $j \in \mathcal{J}$ is allocated to machine $k \in \mathcal{K}_j$ in time-period $t \in \mathcal{T}$; equals 0 otherwise
$z_{jkt} \in \{0, 1\}$	equals 1 if machine $k \in \mathcal{N}_j$ is qualified for job-type $j \in \mathcal{J}$ in time-period $t \in \mathcal{T}$; equals 0 otherwise
$n_t \in \mathbb{R}_+$	maximum resource loading above thresholds ζ_k, $k \in \mathcal{K}$, in time-period $t \in \mathcal{T}$
$\mathbf{y} := (\mathbf{x}, \mathbf{s}, \mathbf{n}, \mathbf{z})$	bold notations representing vectors of the corresponding indexed variables
Parameters	**Description**
$a_{jt} \in \mathbb{Z}_+$	demand of orders of job-type $j \in \mathcal{J}$ in time-period $t \in \mathcal{T}$
$p_{jk} \in \mathbb{Q}_+$	average machining time (including set-up time) in machine $k \in \mathcal{K}_j$ for job-type $j \in \mathcal{J}$
$C_{kt} \in \mathbb{Z}_+$	capacity (hours) available in machine $k \in \mathcal{K}$ in time-period $t \in \mathcal{T}$
$\beta_{jk} \in \mathbb{Z}_+$	nominal qualification cost associated with qualifying machine $k \in \mathcal{N}_j$ for job-type $j \in \mathcal{J}$
$\gamma \in \mathbb{Z}_+$	upper limit on the number of qualifications in a single time-period
$\tau \in \mathbb{Z}_+$	upper limit on number of alternative machines for each job-type in a single time-period
$\zeta_k \in [0, 1]$	loading threshold for machine $k \in \mathcal{K}$

Total qualification cost (g_2) The objective function g_2, to be minimized, is defined as the sum of the one-time costs incurred by qualifying machines for job-types, over all the time-periods, i.e. (2.1b). An increase in the number of qualifications may enable a reduction of the excess loading of the machines.

2.2.1 Model description [Deterministic-TRAP]

The minimization objectives defined in the previous subsection are mathematically expressed as

$$\underset{\mathbf{x},\mathbf{s},\mathbf{n},\mathbf{z}}{\text{minimize}} \qquad g_1(\mathbf{x},\mathbf{s},\mathbf{n},\mathbf{z}) := \sum_{t\in\mathcal{T}} n_t, \tag{2.1a}$$

$$\underset{\mathbf{x},\mathbf{s},\mathbf{n},\mathbf{z}}{\text{minimize}} \qquad g_2(\mathbf{x},\mathbf{s},\mathbf{n},\mathbf{z}) := \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{N}_j} \beta_{jk} z_{jkt}, \tag{2.1b}$$

while the feasible set is described by the constraints

$$\begin{aligned}
\text{s.t.} \quad & \sum_{k\in\mathcal{K}_j} x_{jkt} = a_{jt}, && j\in\mathcal{J},\, t\in\mathcal{T}, && (2.2a)\\
& x_{jkt} \le \min\left\{a_{jt}, \left\lfloor \tfrac{C_{kt}}{p_{jk}} \right\rfloor\right\} s_{jkt}, && k\in\mathcal{K}_j,\, j\in\mathcal{J},\, t\in\mathcal{T}, && (2.2b)\\
& \sum_{k\in\mathcal{K}_j} s_{jkt} \le \tau, && j\in\mathcal{J},\, t\in\mathcal{T}, && (2.2c)\\
& \tfrac{1}{C_{kt}} \sum_{j\in\mathcal{J}} p_{jk} x_{jkt} - \zeta_k \le n_t, && k\in\mathcal{K},\, t\in\mathcal{T}, && (2.2d)\\
& \sum_{l\in\mathcal{T}: l\le t} z_{jkl} \ge s_{jkt}, && k\in\mathcal{N}_j,\, j\in\mathcal{J},\, t\in\mathcal{T}, && (2.2e)\\
& \sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{N}_j} z_{jkt} \le \gamma, && t\in\mathcal{T}, && (2.2f)\\
& x_{jkt} \in \mathbb{Z}_+, && k\in\mathcal{K}_j,\, j\in\mathcal{J},\, t\in\mathcal{T}, && (2.2g)\\
& s_{jkt} \in \{0,1\}, && k\in\mathcal{K}_j,\, j\in\mathcal{J},\, t\in\mathcal{T}, && (2.2h)\\
& z_{jkt} \in \{0,1\}, && k\in\mathcal{N}_j,\, j\in\mathcal{J},\, t\in\mathcal{T}, && (2.2i)\\
& 1-\zeta_k \ge n_t \ge 0, && t\in\mathcal{T}, k\in\mathcal{K}. && (2.2j)
\end{aligned}$$

Defining[1] $\mathbf{y} := (\mathbf{x},\mathbf{s},\mathbf{n},\mathbf{z})$, for any values of $\tau,\gamma \in \mathbb{Z}_+$ the set of feasible

[1] The notations $(\mathbf{x},\mathbf{s},\mathbf{n},\mathbf{z})$ and $\mathbf{y}$ will be used interchangeably throughout this section.

solutions to the model (2.2) is denoted as

$$Y(\tau, \gamma) := \{\, \mathbf{y} \mid \text{the constraints (2.2a)–(2.2j) hold} \,\}. \tag{2.3}$$

We denote the number of orders of job-type $j \in \mathcal{J}$, processed in machine $k \in \mathcal{K}_j$ in time-period t by the decision variables x_{jkt}. In our model, the number of job-types processed should be equal to the demand for each job-type $j \in \mathcal{J}$ in each time-period $t \in \mathcal{T}$, as expressed in (2.2a). The constraints (2.2b) ensure that the number of orders x_{jkt} of job-type j performed in machine k in time-period t does not exceed the demand a_{jt} or available capacity; they also set an auxiliary variable $s_{jkt} = 1$ whenever $x_{jkt} > 0$. The constraints (2.2c) set an upper bound for each job-type and time-period, the number of machines to be used to τ, the value of which is given as an input by the user. The reason behind the use of this constraint is to keep the product flow less complex (see Section 2.1). The constraints (2.2d) make sure to minimize the maximum excess loading above a given threshold for each machine (referred as ζ_k) by setting an upper bound on the variables n_t for each time-period. Furthermore, a binary variable z_{jkt} equals one when a job-type $j \in \mathcal{J}$ is qualified for machine $k \in \mathcal{N}_j$. The constraints (2.2e) imply that if a job-type j is performed in a machine $k \in \mathcal{N}_j$ in time-period t, where $\mathcal{N}_j$ is the set of machines that have *not* been qualified for job-type j, then a qualification of machine k for job-type j must be done once within the time-periods $\{1, \ldots, t\}$. The constraints (2.2f) limit the number of qualifications allowed to be scheduled in each time-period to γ.

The constraints (2.2g), (2.2h), (2.2i), and (2.2j) define the allowed values of the variables x_{jkt}, s_{jkt}, and z_{jkt}, n_t, respectively. The two objectives (2.1a) and (2.1b) represent the sum of excess loading above thresholds and the sum of qualification cost incurred by the planners, respectively. Clearly, this a bi-objective mixed integer programming (BOMIP) model.

2.2.2 Model description [Robust-TRAP]

In **Paper II**, the uncertainty in the qualification cost parameter $\boldsymbol{\beta}$ is considered. Since these are one-time costs (non-repeatable events) we believe it is not appropriate to use a stochastic programming approach as we cannot achieve long-run optimality. Consequently, we reply on a robust optimization approach. It is well-known from the robust optimization literature that when dealing with robust counterparts of a deterministic optimization problem, the selection of an uncertainty set is the most crucial part in hedging against unwanted events (for more details, see [Ben-Tal et al., 2009, Chapter 3]; Bertsimas and Sim [2004]). The two types of uncertainty sets that are commonly used are *finite* *uncertainty sets* and *polyhedral uncertainty sets* (see, [Kuhn et al., 2016, Section 3.2] for definitions).

For the applications considered in this work, the qualification cost of each allocation (j, k), where, $j \in \mathcal{J}$ and $k \in \mathcal{N}_j$ is a natural number; hence, a finite uncertainty set is considered. We define two scenarios, the so-called *nominal-case* (most likely case) and a *worst-case* of qualification cost for each job-type j to be qualified for a machine $k \in \mathcal{N}_j$. It is common in the robust optimization literature to assume a nominal or most likely scenario (see Bertsimas and Sim [2004]). We represent the indices of scenarios by $\mathcal{Q} := \{\hat{q}, \tilde{q}\}^2$, where $\hat{q}$ and $\tilde{q}$ refer to the nominal and the worst-case scenarios, respectively. It is to be noted that the qualification cost in the nominal scenario, i.e. $\beta^{\hat{q}}_{jk}$ is always lower than or equal to that of the worst-case scenario i.e. $\beta^{\tilde{q}}_{jk}$.

Thus, in a robust counterpart to the deterministic TRAP, we can define an objective function $\mathbf{g} : Y(\tau, \gamma) \times \mathcal{Q} \mapsto \mathbb{R}^2_+$, i.e. the scenarios in $\mathcal{Q}$ affect the objective values. Hence, an uncertain bi-objective TRAP is defined as

$$\mathcal{P}(\mathcal{Q}) := \{\mathcal{P}(q),\ q \in \mathcal{Q}\}, \tag{2.4a}$$

$$\tag{2.4b}$$

where $\mathcal{P}(q)$ is defined as

$$\min_{(\mathbf{x},\mathbf{s},\mathbf{n},\mathbf{z}) \in Y(\tau,\gamma)} g((\mathbf{x},\mathbf{s},\mathbf{n},\mathbf{z}), q) := \min_{(\mathbf{x},\mathbf{s},\mathbf{n},\mathbf{z}) \in Y(\tau,\gamma)} \begin{pmatrix} g_1(\mathbf{x},\mathbf{s},\mathbf{n},\mathbf{z}) \\ g_2((\mathbf{x},\mathbf{s},\mathbf{n},\mathbf{z}), q) \end{pmatrix}, \tag{2.4c}$$

and

$$g_2((\mathbf{x},\mathbf{s},\mathbf{n},\mathbf{z}), q) := \sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{N}_j} \beta^q_{jk} z_{jkt}, \qquad q \in \mathcal{Q}, \tag{2.4d}$$

$$g_1(\mathbf{x},\mathbf{s},\mathbf{n},\mathbf{z}) := \sum_{t \in \mathcal{T}} n_t. \tag{2.4e}$$

The concept of *efficient solutions* (see [Miettinen, 1988, Section 2.7]) from multi-objective optimization literature is not entirely valid here. Hence, it is necessary to define alternative concepts that result in desirable solutions when one of the objective functions is uncertain. Paper II deals with the bi-objective robust optimization problem with an uncertain objective function.

[2]More than two scenarios may exist and the methods presented in Paper II can be applied to such uncertainty sets as well

2.2.3 Generalized tactical resource allocation model [GTRAP]

The two variants of the model, Deterministic-TRAP and Robust-TRAP, are explored in Paper I and Paper II, respectively. Furthermore, in Paper III and Paper IV, the consideration is extended to include inventories. In the context of Eq. (2.2a), this implies the presence of a buffer that allows for production levels that may exceed or fall short of demand, as long as the discrepancy is offset by the corresponding inventory. Additionally, rather than treating each machining task as an independent job, we adopt a multi-level series structure for each part type. This approach facilitates the maintenance of inventory for both semi-finished and finished products. We refer to this model variant as the Generalized Tactical Resource Allocation Problem (GTRAP).

3 Related scientific fields

The theoretical background required to solve variants of the TRAP model is presented. The objective functions are linear functions w.r.t. the decision variables and the constraints are affine functions of the same variables. The decision variables are constrained to have one of the following properties: continuous, integer, and binary. Hence, our problems are bi-objective mixed integer linear programming (BOMILP) problems. The latter variant (Robust-TRAP) is also a bi-objective mixed integer linear programming problem (similar to the Deterministic-TRAP) but with an uncertain objective function. A proof of the $\mathcal{NP}$-hardness of the TRAP is presented in Paper I. Hence, solving the TRAP (or any other general problem of which the TRAP is a special case) is computationally hard, especially for the large instances considered in the given industrial problem. Hence it is important that efforts are made to solve a bi-objective MILP as well as a robust bi-objective MILP in a reasonable time-frame. Furthermore, in Paper III we have also introduced a tri-objective variant that has similar (if not more) computational challenges. Ease of interpretation and reasonable computation times are generally extremely important for an optimization model which is part of a decision-making tool. For this purpose, in the coming sections some of the relevant theory that builds the background for the contributions in Papers I, II, III, and IV are discussed.

While an effort has been made to provide an overview of the applicable literature for all four papers within this chapter, it is advised to refer to the individual literature sections of each paper for a comprehensive understanding.

3.1 Preliminaries: MILP and MOMLP

In this section, some preliminaries for Mixed Integer Linear Programming (MILP) problems and Multi-Objective Mixed Integer Linear Programming

(MOMILP) problems are discussed.

3.1.1 Mixed Integer Linear Programming and Solution Methods

A mixed integer linear programming problem is of the form

$$\begin{aligned} \min \quad & \mathbf{c}^\top \mathbf{x} + \mathbf{h}^\top \mathbf{y}, && (3.1a) \\ \text{s.t.} \quad & A\mathbf{x} + G\mathbf{y} \leq \mathbf{b}, && (3.1b) \\ & \mathbf{x} \in \mathbb{Z}_+^{n_1}, && (3.1c) \\ & \mathbf{y} \in \mathbb{R}_+^{n_2}, && (3.1d) \end{aligned}$$

where the data, assumed rational, are denoted as $\mathbf{c} \in \mathbb{Q}_+^{n_1}$, $\mathbf{h} \in \mathbb{Q}_+^{n_2}$, $A \in \mathbb{Q}^{m \times n_1}$, and $G \in \mathbb{Q}^{m \times n_2}$. The decision variable vector $\mathbf{x}$ is non-negative and integral, and the variable vector $\mathbf{y}$ is non-negative and continuous. The feasible set to (3.1) is denoted

$$S := \{(\mathbf{x}, \mathbf{y}) \in \mathbb{Z}_+^{n_1} \times \mathbb{R}_+^{n_2} \mid A\mathbf{x} + G\mathbf{y} \leq \mathbf{b}\},$$

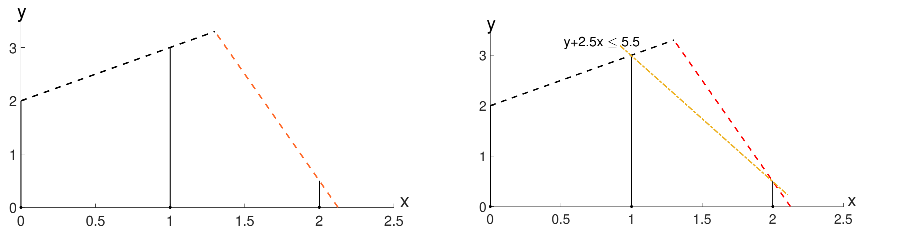
which can be referred to as a mixed integer set. Generally, MILPs are computationally hard to solve, and thus, continuous relaxations of MILPs are extensively used to (hopefully) get good approximations of an optimal solution. The reason is that linear programs (LPs) are generally easier to solve. The natural continuous relaxation of the set S is

$$S_0 := \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}_+^{n_1} \times \mathbb{R}_+^{n_2} \mid A\mathbf{x} + G\mathbf{y} \leq \mathbf{b}\}, \tag{3.2}$$

and the corresponding linear program is $\min\{\mathbf{c}^\top \mathbf{x} + \mathbf{h}^\top \mathbf{y} \mid (\mathbf{x}, \mathbf{y}) \in S_0\}$. In Figure 3.1a, a set of mixed integer points in a polyhedron corresponding to the following MILP is illustrated:

$$\begin{aligned} \min \quad & -5x - 2y, && \\ \text{s.t.} \quad & -x + y \leq 2, && \text{(black-dashed)} \\ & 8x + 2y \leq 17, && \text{(red-dashed)} \\ & x, y \geq 0, && \\ & x \in \mathbb{Z}_+. && \end{aligned}$$

Many real-world decision problems are modeled as MILPs. Hence, it is worthwhile to investigate the computational difficulty of solving such problems to



(a) A mixed integer set w.r.t. (3.1) **(b)** Addition of a cutting plane to the problem

Figure 3.1: Mixed integer set and a cutting plane

(near-)optimality. For a significant majority of real-world problems, the size of the problem instances is large enough to discard the possibility of using enumeration techniques. For instance, in an *assignment problem*, there are n jobs to be performed by n machines. The cost of performing a job is c_{jk}, where $j \in \mathcal{J}$ denotes a job, and $k \in \mathcal{K}$ denotes a machine. The optimization problem is to decide the cheapest way to assign all the jobs to machines (the same job cannot be assigned to two or more machines, and the same machine cannot be assigned to two or more jobs). Since the first job can be assigned to any of the n machines, the second job to any one of the $n-1$ machines, and so on, there is a total of $n!$ possible assignments. It is well-known that $n!$ grows exponentially as a function of n. Hence, enumeration is not possible for an instance with a large value of n. Generally, MILPs and ILPs are $\mathcal{NP}$-hard (i.e. polynomial-time algorithms are not available) (see [Conforti et al., 2014, Chapter 1.3]). However, there are some combinatorial optimization problems for which polynomial-time algorithms are available. This happens when a *perfect formulation* is available or can be easily obtained. The linear system of inequalities $A\mathbf{x} + G\mathbf{y} \leq \mathbf{b}$ results in a perfect formulation of the set $S \subset \mathbb{Z}_+^{n_1} \times \mathbb{R}_+^{n_2}$, if $\text{conv}(S) = \{(\mathbf{x}, \mathbf{y}) \in \mathbb{R}_+^{n_1} \times \mathbb{R}_+^{n_2} \mid A\mathbf{x} + G\mathbf{y} \leq \mathbf{b}\}$. For *pure integer sets*, if the constraint matrix is totally unimodular (TU) [Conforti et al., 2014, Chapter 4.2], then the perfect formulation is available. Perfect formulations are available for some of the classical combinatorial optimization problems such as assignment, shortest path, maximum flow, and bipartite matching. A perfect formulation is also available if the linear system of inequalities has *total dual integrality* (see, [Conforti et al., 2014, Chapter 4.6]). In general, perfect formulations can be made available for all combinatorial optimization problems but for the problems that are $\mathcal{NP}$-hard the number of constraints in a perfect formulation grows exponentially as a function of the number of variables.

Solution methods for MILPs Two common components of most of the exact solution methods for solving MILP problems are the branch-and-bound method and the cutting plane method. In practice, there are many stochastic solution methods, which typically do not provide any (lower) bounds. There are also approximation algorithms that provide bounds on the MIP duality gap. However, in this section we focus on two of the popular solution approaches, i.e. branch-and-bound and cutting plane.

- The branch-and-bound algorithm is a method based on the divide-and-conquer principle. Let us denote an optimal solution to a problem defined in (3.1) by $(\mathbf{x}^*, \mathbf{y}^*)$ and the optimal objective value by z^*. Let us denote the optimal solution and value of the corresponding LP relaxation with feasible set S_0 by $(\mathbf{x}^0, \mathbf{y}^0)$ and z^0, respectively. Since $S \subset S_0$, the inequality $z^0 \leq z^*$ holds. If $\mathbf{x}^0$ is integral then it is implied that $(\mathbf{x}^0, \mathbf{y}^0) \in S$, and $z^* = z^0$. However, usually, at least one of the components of the vector $\mathbf{x}^0$ is fractional. The two main building blocks of the branch-and-bound method, as also highlighted in the name are *branching* and *bounding*. The former, also called *variable branching*, is a procedure in which two or more sub-problems are created by restricting the domain of a variable or a group of variables. Bounding of the objective value is done by solving the LP relaxations of the corresponding sub-problems. This is called *linear programming bounding*. The branch-and-bound algorithm maintains a list of linear programming sub-problems to be solved by relaxing integrality of variables, and also adding constraints on the variables, as $x_j \leq \lfloor x_j \rfloor$ and $x_j \geq \lceil x_j \rceil$, in the respective branches. Each linear programming sub-problem is represented as a node in the branch-and-bound tree. For details, we refer to [Conforti et al., 2014, p. 10]. There are various other modern approaches to branching and bounding implemented in commercial solvers.
- The cutting plane method is the second approach that results in better or tighter re-formulations describing the feasible set S. The main idea is to find an inequality (valid inequality) that cuts off feasible solutions in the relaxed problem which are not present in the set S. An inequality $\boldsymbol{\alpha}^\top \mathbf{u} \leq \beta$ is valid for a set $K \subseteq \mathbb{R}^d$, if it is satisfied for every point $\bar{\mathbf{u}} \in K$, where $\boldsymbol{\alpha}$ and β is a rational vector and scalar, respectively.

  Hence, for the first linear relaxation S_0, look for a valid inequality for the set S, for instance, $\hat{\boldsymbol{a}}^\top \mathbf{x} + \hat{\boldsymbol{g}}^\top \mathbf{y} \leq \hat{b}$, where $\hat{\mathbf{a}} \in \mathbb{Q}^{n_1}$, $\hat{\boldsymbol{g}} \in \mathbb{Q}^{n_2}$, $\hat{b} \in \mathbb{Q}$ such that $\hat{\boldsymbol{a}}^\top \mathbf{x}^0 + \hat{\boldsymbol{g}}^\top \mathbf{y}^0 > b$ holds but $\hat{\mathbf{a}}^\top \mathbf{x} + \hat{\mathbf{g}}^\top \mathbf{y} \leq \hat{b}$ holds for all $\mathbf{x} \in S$ (note that $\mathbf{x}^0$ has fractional components). Hence, the feasible set

  $$S_1 := S_0 \cap \{(\mathbf{x}, \mathbf{y}) \mid \hat{\boldsymbol{a}}^\top \mathbf{x} + \hat{\boldsymbol{g}}^\top \mathbf{y} \leq \hat{b}\}$$

  is smaller than S_0. Thus, it is implied that $S \subseteq S_1 \subset S_0$, and the formulation corresponding to the LP relaxation of S_1 is stronger than for the set

S_0. Consequently, $z_0 \leq z_1$, where $z_1 = \min_{(\boldsymbol{x},\boldsymbol{y}) \in S_1} \{\boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{h}^\top \boldsymbol{y}\}$. Note that for a minimization problem it is important to compute as large a lower bound as possible for faster convergence to an optimal solution. For detailed steps the reader should refer to [Conforti et al., 2014, p. 10], and for details on cutting plane methods, the introduction section in [Conforti et al., 2014, Chapter 5] is relevant.

Generally, in most of the commercial solvers, both of these methods are combined in a branch-and-cut framework. In this approach, cuts are added to get tighter formulations before applying branching. For instance, in the example in Figure 3.1a, one of the optimal solutions for the LP relaxation corresponding to S_0 is $(x, y) = (1.3, 3.3)$, and if the constraint $2.5x + y \leq 5.5$ is added, it results in a tighter formulation (see Figure 3.1b). In fact, if the LP relaxation is solved after adding this constraint, a solution $(x, y) = (1, 3)^\top$ is obtained, which is a feasible, and optimal solution to the MILP problem in (3.1).

For most of our work, commercial solvers are used that have advanced/mature sub-routines to generate appropriate cuts and selection rules for branching decisions depending on the type of problem instances. Some of the cuts that are applied are knapsack covers (see Crowder et al. [1983]), GUB covers (see Gu et al. [1999]), flow covers (see Gu et al. [1999]), cliques (see Crowder et al. [1983]), implied bounds (see Hoffman and Padberg [1991]) and Gormory mixed-integer cuts (see Cornuéjols [2006]. There are also various *lifting procedures* (see [Conforti et al., 2014, Chapter 7]) which are used extensively in almost all modern implementations. For more details on commercial codes of solvers, readers should refer to Bixby et al. [2000].

3.1.2 Multi-Objective Mixed Integer Linear Programming Problems

Most industrial decision problems have several objectives, which are often in conflict. Let us consider a multi-objective mixed integer linear programming (MOMILP) problem defined as

$$\min_{\mathbf{x} \in X} \boldsymbol{z}(\mathbf{x}) := (z_1(\mathbf{x}), \ldots, z_p(\mathbf{x})), \tag{3.4}$$

here $X \subseteq \mathbb{Z}^n_+$ is defined by a set of affine constraints, with $\mathbf{x} \in \mathbb{Z}^{n_1}_+ \times \mathbb{R}^{n_2}_+$ variables, where $n = n_1 + n_2$. The functions $z_1, z_2, \ldots, z_p$ are linear, and the image $\mathcal{Z}$ of X under vector valued functions $\boldsymbol{z} : \mathbb{Z}^n_+ \to \mathbb{R}^p_+$ represents the feasible set in the criterion space. Some common notations and definitions used to solve multi-objective optimization problems and define relevant optimality concepts are described next. For any two vectors $\mathbf{z}$ and $\mathbf{w}$, both belonging to $\mathbb{R}^p$ with $p \geq 2$, we employ the following criteria for comparisons:

$$\mathbf{z} \leq \mathbf{w} \quad \iff \quad w_i \in [z_i, \infty) \quad \forall i \in \{1, \dots, p\}; \tag{3.5a}$$

$$\mathbf{z} \preceq \mathbf{w} \quad \iff \quad w_i \in [z_i, \infty) \quad \forall i \in \{1, \dots, p\} \quad \text{and} \quad \mathbf{z} \neq \mathbf{w}; \tag{3.5b}$$

$$\mathbf{z} < \mathbf{w} \quad \iff \quad w_i \in (z_i, \infty) \quad \forall i \in \{1, \dots, p\}. \tag{3.5c}$$

In MOMILPs, it is often the case that a singular optimal value does not exist due to the inherent trade-offs between conflicting objectives. Hence, one must present a set of solutions that represent a balance among the competing objectives. These solutions can then be presented to decision-makers, who can make informed choices based on their specific preferences and the solutions obtained.

Definition 3.1 (Weakly efficient solutions). *A feasible solution* $\mathbf{x}' \in X$ *is called the* weakly efficient solution *if* $\nexists \mathbf{x} \in X$ *such that,* $z_k(\mathbf{x}) < z_k(\mathbf{x}')$*, for* $k \in \{1, \dots, p\}$*. Furthermore,* $\boldsymbol{z}(\mathbf{x}')$ *is called a* weakly non-dominated point in the criterion space*.*

Definition 3.2 (Efficient solutions). *A feasible solution* $\mathbf{x}' \in X$ *is called the* efficient solution *or Pareto optimal solution if* $\nexists \mathbf{x} \in X$ *such that* $\boldsymbol{z}(\mathbf{x}) \preceq \boldsymbol{z}(\mathbf{x}')$*. Furthermore,* $\boldsymbol{z}(\mathbf{x}')$ *is called a* non-dominated point in the criterion space*. The set of all the non-dominated points is called the* efficient frontier*. The set of efficient solutions is denoted by* X_{eff}*. The set of corresponding objective values is called a set of* non-dominated points *(NDPs).*

Definition 3.3 (Ideal point). *A point* $\mathbf{z}^{\text{ideal}} \in \mathbb{R}^p$ *is called an* ideal point *(see [Miettinen, 1988, Definition 2.4.1]) if it minimizes all the objectives separately/individually. Thus,* $z_k^{\text{ideal}} := \min_{\mathbf{x} \in X} z_k(\mathbf{x})$*,* $k \in \{1, \dots, p\}$*.*

Definition 3.4 (Supported efficient solution). *A feasible solution* $\mathbf{x}' \in X_{\text{eff}}$ *is called a* supported efficient *solution (see [Ehrgott, 2005, Definition 8.7]) if* $\exists \boldsymbol{\lambda} > \mathbf{0}^p$ *such that* $\mathbf{x}' \in \arg\min_{\mathbf{x} \in X_{\text{eff}}} \boldsymbol{\lambda}^\top \boldsymbol{z}(\mathbf{x})$ *and* $\boldsymbol{z}(\mathbf{x}')$ *is* supported non-dominated point*. On the contrary, if an efficient solution does not satisfy this condition then it is called a un-supported efficient solution*[1]*.*

Most Multi-Objective Optimization Problems (MOOPs) are tackled using a method known as scalarization. This method involves converting a MOOP into a single-objective optimization problem, which can then be solved using well-established techniques for single-objective optimization. The solution thus obtained serves as a boundary for the subsequent scalarization iteration. This iterative process continues until all non-dominated points have been identified or

[1] Note that un-supported efficient solutions generally exist for non-convex problems

an acceptable approximation is achieved. It is important to note that MOMILP poses a particularly daunting challenge due to the inherent complexity of solving scalarized problems in the form of mixed-integer programs. Nonetheless, MOMILP is a well-researched field, and several developed methods will be discussed in the following section. In the context of this discussion, we are specifically considering problems with only affine functions. However, there is also significant research focused on multi-objective mixed-integer convex (and non-convex) optimization, where multiple convex (or non-convex) objective functions are present and some variables are constrained to integer values. Notable work in this area includes Eichfelder and Warnow [2023], Niebling and Eichfelder [2019]. However, the focus of the literature survey is mainly multi-objective integer linear programming (MOILP) problems as we have also shown that our models have only a discrete number of non-dominated points in Paper I.

3.2 Methods for Solving Multi-Objective Integer Linear Programming (MOILP) Problems

Algorithms for *Multi-Objective Integer Linear Programming* (MOILP) can be broadly classified into two main categories: *decision space search methods* and *criterion space search methods*.

Popular methods for decision space search include evolutionary multi-objective methods, such as NSGA-II (see Deb et al. [2002]), which has gained interest, although it does not provide any measure of the verified closeness to the Pareto front. There have been some improvements suggested in branch-and-bound methods for mixed 0-1 linear problems (e.g. Vincent et al. [2013] and Stidsen et al. [2014]), which is a decision space search method.

Our work focuses on criterion space search methods, that provide (approximate) efficient frontiers, and which are also motivated by an improved efficiency of mathematical optimization solvers and relatively inexpensive computing power. Some of the popular methods for criterion space search are the *weighted sum method* (e.g. Aneja and Nair [1979]), the *perpendicular search method* (see Chalmet et al. [1986]), the *augmented weighted Tchebycheff* (AWT) method (e.g. Bowman [1976] and Steuer and Choo [1983]), and the *ϵ-constraint method* (see [Miettinen, 1988, p. 85]). Most of the algorithms suggested in the literature have one basic operation common among them, the so-called *scalarization*. The idea is to transform a MOILP into a series of single-objective optimization problems which are solved sequentially.

Definition 3.5 (Scalarized problem)**.** *A scalarized problem is a single-objective optimization problem related to MOILP with additional variables, and constraints solved repeatedly in order to find some subset of the set of efficient solutions (see Ehrgott [2006]).*

The two main aspects considered while choosing a scalarization are (a) Is an optimal solution of the scalarized problem a weakly or strictly efficient solution? (b) Can all the efficient solutions be identified (both supported and un-supported efficient solutions)? Following are some of the popular scalarization techniques used to identify efficient solutions:

- *Weighted Sum method*: This is one of the most popular methods for solving both MOILPs and MOMILPs. In this method, each objective function is associated with a non-negative coefficient, and hence, transformed into a single-objective optimization problem. The following model is a typical representation of the scalarization used in the weighted sum method:

$$\min_{\mathbf{x}\in X} \sum_{k=1}^{p} \lambda_k \mathbf{c}_k^\top \mathbf{x}, \tag{3.6}$$

  where $\lambda_k > 0$ is the weight coefficient for each objective function indexed by $k \in \{1,\ldots,p\}$. The cost coefficient vector for the k^{th} objective function is $\mathbf{c}_k \in \mathbb{R}^n_+$. It is a well-known result (see [Ehrgott, 2005, Chapter 3]) that any solution to the model (3.6) is an efficient solution, however, it is always a supported efficient solution. Hence, un-supported efficient solutions are not identified by the weighted sum method. One important advantage of the weighted sum method is that it (model (3.6) for a given $\boldsymbol{\lambda}$) usually requires the same computational effort as a single-objective version of the MOILP or MOMILP.

- *ϵ-constraint method*: It is a popular type of method capable of finding all the efficient solutions (both supported as well as un-supported). In this method, only one of the p objective functions is considered and the remaining $p-1$ are set as constraints (also popularly referred to as ϵ-bounds) on the values of the respective objective functions. The values of $\boldsymbol{\epsilon} \in \mathbb{R}^{p-1}$ are updated after each scalarized problem

$$\min_{\mathbf{x}\in X} \quad \mathbf{c}_j^\top \mathbf{x}, \tag{3.7a}$$

$$\text{s.t.} \quad \mathbf{c}_k^\top \mathbf{x} \leq \epsilon_k, \quad k \in \{1,\ldots,p\} \setminus \{j\}, \tag{3.7b}$$

  is solved to optimality. The optimal solution of the model (3.7) is at least weakly efficient, and under certain conditions even strictly efficient (see

Chankong and Haimes [1983] for other results). One of the drawbacks of the ϵ-constraint method is that the scalarized model (3.7) is generally computationally harder as compared to the single objective version. This is mainly due to the constraints (3.7b), which are actually knapsack constraints, added to the problem. For certain types of problems depending on the structure of set X, these additional constraints may make the problem computationally very hard (see [Ehrgott, 2006, Sec 4.4] for specific problem instances).

- *Augmented weighted Tchebycheff (AWT) method*: This method first proposed in Steuer and Choo [1983] is quite popular within *interactive methods* (see [Miettinen, 1988, Chapter 5] for more details) as well. The method adds to the objective function a weighted distance from a reference point (usually an ideal point, $\mathbf{z}^{\text{ideal}} \in \mathbb{R}^p$) in the criterion space. The following model is a typical scalarization used for this purpose

$$\min_{\mathbf{x}\in X} \quad \left\{ f + \bar{\lambda} \sum_{k=1}^{p} \left(z_k(\mathbf{x}) - z_k^{\text{ideal}} \right) \right\}, \tag{3.8a}$$

$$\text{s.t.} \quad f \geq \alpha_k \left(z_k(\mathbf{x}) - z_k^{\text{ideal}} \right), \qquad k \in \{1, \dots, p\} \tag{3.8b}$$

$$f \geq 0, \tag{3.8c}$$

where $\alpha_k > 0$ are the respective weights for the l_∞-norm of the difference between the ideal point ($\mathbf{z}^{\text{ideal}}$) and the objective vector $\mathbf{z}(\mathbf{x})$ corresponding to a point $\mathbf{x} \in X$, and $\bar{\lambda}$ is the coefficient for the l_1-norm of the same distance measure. By choosing appropriate values of $\bar{\lambda}$ and $\boldsymbol{\alpha}$, all non-dominated points can be obtained. The inclusion of min-max objective results in some increased computation time as compared to single-objective MOILPs. Furthermore, identifying $\bar{\lambda}$ and $\boldsymbol{\alpha}$ for searching only strictly efficient or at least fewer weakly efficient solutions has made the use of this method elusive.

- *Benson's method*: First presented in Benson [1978], it is a method that can be used for checking whether a given solution is efficient, and also identifying yet unknown non-dominated points. The scalarized problem can be defined as (note the additional variable vector $\boldsymbol{u}$)

$$\max \sum_{k=1}^{p} u_k, \tag{3.9a}$$

$$\text{s.t.} \quad \mathbf{c}_k^\top \bar{\mathbf{x}} - u_k - \mathbf{c}_k^\top \mathbf{x} = 0, \qquad k = 1, \dots, p, \tag{3.9b}$$

$$\mathbf{u} \geq 0, \tag{3.9c}$$

$$\mathbf{x} \in X, \tag{3.9d}$$

where $\bar{\mathbf{x}} \in X$ is the solution that needs to be checked if it is efficient or not. Let us denote $u_k = c_k^\top \bar{\mathbf{x}} - c_k^\top \mathbf{x}$, a formulation almost a union of the weighted sum and ϵ-constraint method is obtained as

$$\min_{\mathbf{x}\in X} \left\{ \sum_{k=1}^{p} \mathbf{c}_k^\top \mathbf{x} \ : \ \mathbf{c}_k^\top \mathbf{x} \leq \mathbf{c}_k^\top \bar{\mathbf{x}},\ k \in \{1, \ldots, p\} \right\}. \tag{3.10}$$

A general framework suggested by Ehrgott [2006], for scalarized problems is:

$$\min_{\mathbf{x}\in X} \max_{k\in\{1,\ldots,p\}} \left\{ \alpha_k(\mathbf{c}_k^\top \mathbf{x} - \rho_k) + \sum_{k=1}^{p} \lambda_k(\mathbf{c}_k^\top \mathbf{x} - \rho_k) \right\}, \tag{3.11a}$$

$$\text{s.t.} \quad \mathbf{c}_k^\top \mathbf{x} \leq \ \epsilon_k, \qquad k \in \{1, \ldots, p\}, \tag{3.11b}$$

where ρ_k, and α_k, $k \in \{1, \ldots, p\}$, are defined in Table 3.1.

Table 3.1: Parameters for the generalized scalarized problem (3.11), where $\boldsymbol{z}^{\text{ideal}}$ and $\bar{x}$ denote the ideal objective value, and (pre-defined) reference solution, respectively.

Method	ρ_k	α_k	$\boldsymbol{\lambda}$	$\boldsymbol{\epsilon}$
weighted sum	0	0	$\boldsymbol{\lambda} \in \mathbb{R}^p_>$	$\epsilon_k = \infty, k \in \{1, \ldots, p\}$
ϵ-constraint	0	0	$\lambda_j = 1, \lambda_k = 0, k \neq j$	$\epsilon_j = \infty, \epsilon_k \in \mathbb{R}, k \neq j$
AWT	$\mathbf{z}_k^{\text{ideal}}$	> 0	$[\lambda_k]_{k=\{1,\ldots,p\}} = \bar{\lambda} \geq 0$	$\epsilon_k = \infty, k \in \{1, \ldots, p\}$
Benson's	0	0	$[\lambda_k]_{k=\{1,\ldots,p\}} = 1$	$\epsilon_k = \mathbf{c}_k^\top \bar{\mathbf{x}}, k \in \{1, \ldots, p\}$

Another common strategy while looking for efficient solutions is the so-called *two-phase strategy*. Generally, in the first phase, all the supported efficient solutions are identified, whereas in the second phase, un-supported efficient solutions are identified. This method was used to solve a bi-objective assignment problem in Przybylski et al. [2008]. However, these two-phase approaches have appeared previously as well (see Visée et al. [1998]). Some other methods that have been proposed for MOILP problems can be reviewed in [Ehrgott, 2006, Sec. 2].

3.3 Specific algorithms for BOILP and TOILP problems

In Paper I, a bi-objective mixed integer linear programming (BOMILP) problem called Tactical Resource Allocation Problem (TRAP) is solved. An important conclusion drawn from Proposition 3 in Paper I is the following

Proposition 3.1 (Efficient frontier of the TRAP model)**.** *The efficient frontier of the TRAP contains only isolated non-dominated points, and no (closed, half open, or open) line segments (as is the case in typical BOMILPs), irrespective of the values of the parameters β_{jk}, $k \in \mathcal{N}_j$, $j \in \mathcal{J}$.*

Hence, without loss of generality, one can use algorithms for bi-objective integer linear programming (BOILP) problems for solving the TRAP as well. Most of the algorithms mentioned in the previous section are popular while solving BOILPs. However, some algorithms based on decomposing the criterion space also exist. One of the important (recent) ones, called the *Balanced Box* method, is designed for BOIPs (see Boland et al. [2015] for details). Generally, in these criterion space decomposition methods, more number of scalarized problems are solved as compared to most of the conventional scalarization methods but usually, these scalarized problems in decomposition methods are computationally easier to solve. In this section, we have also introduced a specialized method for the tri-objective integer linear programming (TOILP) problem which is useful in Paper III and Paper IV.

3.3.1 Balanced Box method

Balanced Box method is used to identify efficient frontier of bi-objective integer programming problems. There is an initial search space defined by the two non-dominated points $\mathbf{z}^{\mathrm{T}}$ and $\mathbf{z}^{\mathrm{B}}$, which refer to the non-dominated points defining the minimum value for the first and second objective functions, respectively. Formally, the two non-dominated points are defined as

$$\mathbf{z}^{\mathrm{T}} := \operatorname*{lexmin}_{\mathbf{x}\in X}\{z_1(\mathbf{x}), z_2(\mathbf{x})\}, \tag{3.12a}$$

$$\mathbf{z}^{\mathrm{B}} := \operatorname*{lexmin}_{\mathbf{x}\in X}\{z_2(\mathbf{x}), z_1(\mathbf{x})\}, \tag{3.12b}$$

where lexmin is the standard lexicographic minimization as defined in [Ehrgott, 2005, Section 5.1].

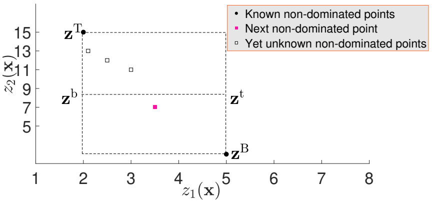


Figure 3.2: First step of the Balanced Box method

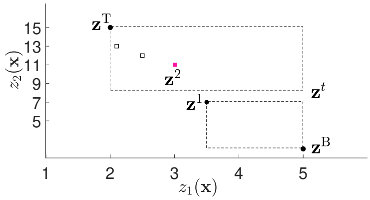


Figure 3.3: Second step of the Balanced Box method (legends as Figure 3.2)

The rectangular search space is then defined as

$$\mathrm{R}(\mathbf{z}^{\mathrm{T}}, \mathbf{z}^{\mathrm{B}}) := \left\{ \mathbf{z} \in \mathbb{R}^2 \,\middle|\, z_1^{\mathrm{T}} \leq z_1 \leq z_1^{\mathrm{B}},\ z_2^{\mathrm{B}} \leq z_2 \leq z_2^{\mathrm{T}} \right\},$$

where $(z_1^{\mathrm{T}}, z_2^{\mathrm{B}})$ and $(z_1^{\mathrm{B}}, z_2^{\mathrm{T}})$ denote the *ideal* and *nadir points*, respectively (see [Miettinen, 1988, p. 15–16]). In Figures 3.2 and 3.3, a simple example of this procedure is illustrated. In the first step (see Figure 3.2), there are two initial non-dominated points ($\mathbf{z}^{\mathrm{T}}$ and $\mathbf{z}^{\mathrm{B}}$). Furthermore, the rectangle $\mathrm{R}(\mathbf{z}^{\mathrm{T}}, \mathbf{z}^{\mathrm{B}})$ is split into two halves along the z_2 axis. Thus, the two new rectangles containing yet-unknown non-dominated points are $\mathrm{R}(\mathbf{z}^{\mathrm{T}}, \mathbf{z}^{\mathrm{t}})$ and $\mathrm{R}(\mathbf{z}^{\mathrm{b}}, \mathbf{z}^{\mathrm{B}})$, where $\mathbf{z}^b := (z_1^{\mathrm{T}}, \frac{z_2^{\mathrm{T}}+z_2^{\mathrm{B}}}{2})$ and $\mathbf{z}^t := (z_1^{\mathrm{B}}, \frac{z_2^{\mathrm{T}}+z_2^{\mathrm{B}}}{2})$. Firstly, the rectangle $\mathrm{R}(\mathbf{z}^{\mathrm{b}}, \mathbf{z}^{\mathrm{B}})$ is searched and the problem $\operatorname*{lexmin}_{\mathbf{x}\in X}\{z_1(\mathbf{x}), z_2(\mathbf{x}) \mid \mathbf{z} \in \mathrm{R}(\mathbf{z}^{\mathrm{b}}, \mathbf{z}^{\mathrm{B}})\}$ is solved. The non-dominated point obtained is $\mathbf{z}^1$ (see illustration in Figure 3.3). Similarly, a lexicographic minimization problem is solved for the other rectangle $\mathrm{R}(\mathbf{z}^{\mathrm{T}}, \mathbf{z}^{\mathrm{t}})$ to find $\mathbf{z}^2$ using $\operatorname*{lexmin}_{\mathbf{x}\in X}\{z_2(\mathbf{x}), z_1(\mathbf{x}) \mid \mathbf{z} \in \mathrm{R}(\mathbf{z}^{\mathrm{T}}, \mathbf{z}^{\mathrm{t}})\}$. A recursive algorithm is presented in [Boland et al., 2015, Algorithm 2]; this algorithm has shown computational superiority over many existing methods for several benchmarking instances for BOIPs (see [Boland et al., 2015, Section 6]).

3.3.2 AWT (with adaptive formulae)

The Augmented Weighted Tchebycheff (AWT) method is discussed in Section 3.2. As already highlighted, one of the issues with the AWT method is, however, that the coefficients for the l_∞- and l_1-norms are not available; hence there is a risk that many weakly efficient solutions are identified[2]. In Dächert et al. [2012] the authors came up with adaptive formulae for solving bi-objective integer linear programming problems and the summary is in Table 3.2. For a given reference point in the criterion space $\mathbf{z}^{\mathrm{ideal}}$, define

[2]sometimes using too small coefficients also result in numerical insatiability.

$x := z_1^{\mathrm{B}} - z_1^{\mathrm{ideal}}, y := z_2^{\mathrm{T}} - z_2^{\mathrm{ideal}}$, and $u \in (0, 1)$. Note that when $x \leq 1$, or $y \leq 1$ (not considered in Table 3.2), since it is a BOILP, there are no interior non-dominated points in $R(\mathbf{z}^{\mathrm{T}}, \mathbf{z}^{\mathrm{B}})$, where $\mathbf{z}^{\mathrm{T}}$ and $\mathbf{z}^{\mathrm{B}}$ are described in (3.12).

Table 3.2: Parameters for the AWT method (3.8) for BOILPs from [Dächert et al., 2012, Table 2]

Case	α_1	α_2	$\bar{\lambda}$
$x > y \geq 2$	$\frac{xy-x-y+u(2-u)}{xy-y-3x+x^2+2u(2-u)}$	$\frac{(x-u)(x+u-2)}{xy-y-3x+x^2+2u(2-u)}$	$\frac{(x-u)(1-u)}{xy-y-3x+x^2+2u(2-u)}$
$x = y \geq 2$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1-u}{2(x+u-2)}$
$y > x \geq 2$	$\frac{(y-u)(y+u-2)}{xy-x-3y+y^2+2u(2-u)}$	$\frac{xy-x-y+u(2-u)}{xy-x-3y+y^2+2u(2-u)}$	$\frac{(y-u)(1-u)}{xy-x-3y+y^2+2u(2-u)}$

3.3.3 Quadrant Shrinking method (for TOILP problems)

There are numerous specialized algorithms described in the literature specifically designed for solving MOILPs (see some popular ones Sylva and Crema [2004]; Dächert et al. [2017]; Lokman and Köksalan [2012]), and for a detailed review, we refer to [Boland et al., 2017, Section 1]. One of the latest additions to algorithms for tri-objective integer linear programming (TOILP) problems is the Quadrant Shrinking Method (QSM) presented in Boland et al. [2017]. Our interest in TOILP problems is due to Paper III which introduces a generalized tactical resource allocation problem (GTRAP). The QSM forms the basis of our decomposition approaches in both Papers III and IV. The reason behind the selection of the QSM is that it operated on the projected two-dimensional criterion space that facilitated our decomposition approach.

A TOILP can be described as in (3.4), with $p = 3$. QSM works in projected two-dimensional criterion space. A point $\boldsymbol{u} := (u_1, u_2, u_3)^\top$ is projected as $\hat{\boldsymbol{u}} = (u_1, u_2)$ in the criterion space corresponding to z_1 and z_2. Given $\hat{\mathbf{u}} \in \mathbb{R}^2$, a quadrant is defined as $Q(\hat{\mathbf{u}}) := \{\mathbf{y} \in \mathbb{R}^2 \mid \mathbf{y} \leq \hat{\mathbf{u}}\}$, hence, $\hat{\mathbf{u}}$ is the upper bound of the quadrant $Q(\hat{\mathbf{u}})$. As a result of [Boland et al., 2017, Propositions 4 and 5], it is established that a non-dominated point $\mathbf{z}(\hat{\mathbf{x}})$, with the property that its projection $(z_1(\hat{\mathbf{x}}), z_2(\hat{\mathbf{x}}))^\top \leq \hat{\mathbf{u}}$, can be found by solving the following two ILPs

$$\mathbf{x}^* \in \arg\min_{\mathbf{x}} \{z_3(\mathbf{x}) : \mathbf{x} \in X; z_k(\mathbf{x}) \leq \hat{u}_k, k \in \{1, 2\}\}, \tag{3.13a}$$

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \left\{ \sum_{k=1}^{3} z_k(\mathbf{x}) : \mathbf{x} \in X; z_3(\mathbf{x}) \leq z_3(\mathbf{x}^*); z_k(\mathbf{x}) \leq \hat{u}_k, k \in \{1, 2\} \right\}. \tag{3.13b}$$

This is referred to as *2-D-NDP search* in Boland et al. [2017], however, originally it first appeared as a *two-stage scalarization* in Kirlik and Sayın [2014]. This is the core step in QSM used to explore all the quadrants that are expected to have yet unknown non-dominated points. A recursive algorithm is detailed in [Boland et al., 2017, Algorithm 1].

3.4 Uncertainty in the objective functions

The multi-objective optimization problem (3.4) has no uncertainty associated with parameters. Sometimes uncertainty present in parameters may arise due to uncertain future developments of the data defining an instance, and some imprecise calculations or measurements. The outcome of decisions made under uncertainty of some parameters can sometimes be extremely sensitive to the actual data, and hence, extra care should be taken while making decisions under uncertainty.

For this particular reason, different approaches have been suggested for solving MOOPs which are based on stochastic programming, fuzzy approaches, and robust optimization. Stochastic programming for MOOP (see Gutjahr and Pichler [2013]) is used when there are enough data available and fuzzy approaches (see [Slowinski and Teghem, 1990, Chapter 4]) when expert judgments on fuzzy membership are reliable. A drawback of the stochastic approach is that for some problems so-called *long-run optimality* is not relevant, as the *repeatability element* of the decisions is missing; the decision maker has to live with the consequences of the decision made once. Since in TRAP, qualification costs for a specific type of allocation is incurred only once, it is evident that combining robust optimization and multi-objective optimization has certain benefits over other approaches.

Recently, the robust multi-objective optimization approach has been gaining interest in the research community for solving multi-objective optimization problems with uncertain objective functions, and deterministic constraints, for which the amount of data available is not sufficient to make any informed probability distribution assumptions of the input parameters. An uncertain MOOP (with deterministic constraints) can be defined as a family of parameterized problems as follows:

$$\mathcal{P}(\mathcal{U}) := \ (\mathcal{P}(\boldsymbol{\xi}), \boldsymbol{\xi} \in \mathcal{U}), \tag{3.14a}$$

where $P(\xi)$ is defined as

$$\begin{aligned} \min \quad & \mathbf{z}(\mathbf{x}, \boldsymbol{\xi}), \qquad \boldsymbol{\xi} \in \mathcal{U}, && (3.15a) \\ \text{s.t.} \quad & \mathbf{x} \in X, && (3.15b) \end{aligned}$$

where $\mathbf{z} : X \times \mathcal{U} \rightarrow \mathbb{R}^p$, $\boldsymbol{\xi}$ is a vector containing uncertain parameters and $\mathcal{U}$ is the set of uncertain scenarios. There are two main types of uncertainty sets considered in the robust optimization literature:

- **Finite uncertainty set**. In this case, it is assumed that the set of scenarios is finite, i.e. $\mathcal{U} = \{\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^n\}$.
- **Polyhederal uncertainty**. The uncertainty set is given as the convex hull of a finite set of scenarios, i.e. $\mathcal{U} = \text{conv}\{\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^n\}$.

Next, single-objective robust optimization and its generalization to the multi-objective case is introduced.

3.4.1 Single objective robust optimization

For single objective robust optimization (SO-RO) problems with deterministic constraints, numerous concepts of robustness have been discussed in the literature. Some of the well-known ones are:

- *Minmax robust optimality for SO-RO problems* (see [Ide and Schöbel, 2016, Definition 11]). Given $P(\mathcal{U})$ with $\mathbf{z} : X \times \mathcal{U} \rightarrow \mathbb{R}$ (i.e only one objective function), a solution is called minmax robust optimal if it is an optimal solution to
$$\min_{\mathbf{x} \in X} \max_{\boldsymbol{\xi} \in \mathcal{U}} \{ \ \mathbf{z}(\mathbf{x}, \boldsymbol{\xi}) \ \}.$$
- *Minmax regret* is a concept to avoid conservativeness of the minmax approach. *Regret* is defined as the difference between the resulting benefit (cost) to the decision maker, and the benefit (cost) to the decision maker from the decision if the actual scenario was known (see [Kouvelis and

Yu, 1997, Chapter 1] for details). There are many other variants of regret as well, one of them being the relative deviation of the objective value corresponding to the robust decision from the value corresponding to the optimal decision if the actual scenario is known.

- *Light robustness* is introduced for SO-RO in Fischetti and Monaci [2009]. The idea is to choose solutions that are considered ε "good enough" in the nominal (most likely) scenario and select the one that is most reliable in the worst-case scenario within a certain interval. This approach also reduces the over-conservativeness of the minmax approach which is a common criticism of robust optimization as well.

A recent review article on SO-RO is Goerigk and Schöbel [2016].

3.4.2 Robust MOOP

The need to develop efficiency concepts for robust MOOP first arose due to requirements in certain application areas of aircraft route guidance and shipping hazardous materials (see, [Kuhn et al., 2016, Section 8] for more details on applications). Defining an analogous concept of efficient solutions (from MOOP) to a comparable concept in robust MOOP is not straightforward. Various concepts of the so-called *robust efficiency* have been suggested. A detailed overview of various robust efficiency concepts is presented in [Ide and Schöbel, 2016, Section 3]. Following are some of the robust efficiency concepts that are important for Paper II

Definition 3.6 (*Flimsily robust efficient* (FRE))**.** *Given the uncertain MOOP $\mathcal{P}(\mathcal{U})$, a solution $\bar{\mathbf{x}} \in X$ is called* flimsily robust efficient *(FRE) for $\mathcal{P}(\mathcal{U})$ if it is efficient for $\mathcal{P}(\boldsymbol{\xi})$ for at least one $\boldsymbol{\xi} \in \mathcal{U}$. The set FRE solutions $X^f := \bigcup\limits_{\boldsymbol{\xi}\in\mathcal{U}} X_{\text{eff}}(\boldsymbol{\xi})$, where $X_{\text{eff}}(\boldsymbol{\xi})$, is the set of efficient solutions to the deterministic MOOP $\mathcal{P}(\boldsymbol{\xi})$.*

Definition 3.7 (*Highly robust efficient* (HRE))**.** *Given the uncertain MOOP $\mathcal{P}(\mathcal{U})$, a solution $\bar{\mathbf{x}} \in X$ is called* highly robust efficient *(HRE) for $\mathcal{P}(\mathcal{U})$ if it is efficient for $\mathcal{P}(\boldsymbol{\xi})$ for all $\boldsymbol{\xi} \in \mathcal{U}$. The set of HRE solutions $X^h := \bigcap\limits_{\boldsymbol{\xi}\in\mathcal{U}} X_{\text{eff}}(\boldsymbol{\xi})$.*

Remark 3.1. *For the two special cases following holds*

- *For $|\mathcal{U}| = 1$ (i.e. MOOP), the set of HRE and FRE solutions are equivalent.*
- *For $p = 1$ (one objective function, i.e. single-objective robust optimization), a solution is HRE if it is optimal for all the scenarios $\boldsymbol{\xi} \in \mathcal{U}$ and FRE if it is optimal to at least one of the scenarios.*

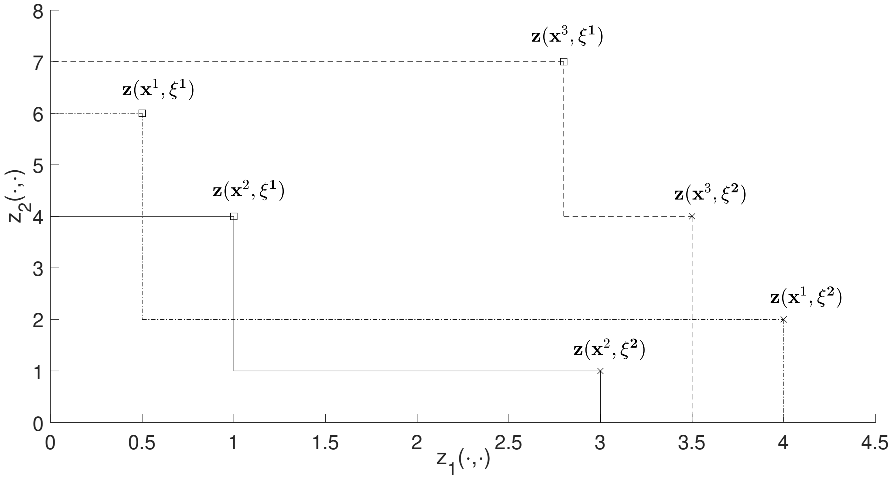


Figure 3.4: $\mathbf{z}_{\mathcal{U}}(\mathbf{x}^1) - \mathbb{R}^2_{\succeq}$ (dash-dotted), $\mathbf{z}_{\mathcal{U}}(\mathbf{x}^2) - \mathbb{R}^2_{\succeq}$ (solid), $\mathbf{z}_{\mathcal{U}}(\mathbf{x}^3) - \mathbb{R}^2_{\succeq}$ (dashed). The cross and square-marks are for scenarios $\boldsymbol{\xi}^1$ and $\boldsymbol{\xi}^2$, respectively.

HRE is a very restrictive requirement, and the existence of such solutions is not guaranteed. However, as per [Ide and Schöbel, 2016, Lemma 9], the existence of such a solution (i.e. HRE) is guaranteed, if one of the objectives does not have any uncertain parameters i.e. at least one of the objective function $i^* \in \{i \in \{1, \dots, p\}; z_i(\mathbf{x}, \boldsymbol{\xi}') = z_i(\mathbf{x}, \bar{\boldsymbol{\xi}}), \boldsymbol{\xi}', \bar{\boldsymbol{\xi}} \in \mathcal{U}\}$ *exists, and also has a unique optimal solution to the problem* $\min\{z_{i^*}(\mathbf{x}, \cdot) \mid \mathbf{x} \in X\}$[3].

For SO-RO problems, minmax robust optimality is well-defined but when there is a vector valued objective function, the definition of the worst-case is not unambiguous. Hence, an extension of minmax robustness to MOOP is not unambiguously defined. There are three extensions of this concept for MOOP. The most common one—from Ehrgott et al. [2014]—is presented next.

Definition 3.8 (Set-based minmax robust efficiency, Ehrgott et al. [2014])**.** *Given the uncertain MOOP $\mathcal{P}(\mathcal{U})$, a feasible solution $\bar{\boldsymbol{x}} \in X$ is called* set-based minimax RE solution *if $\nexists \mathbf{x}' \in X \setminus \{\bar{\mathbf{x}}\}$, such that*

$$\mathbf{z}_{\mathcal{U}}(\mathbf{x}') \subseteq \mathbf{z}_{\mathcal{U}}(\bar{\mathbf{x}}) - \mathbb{R}^p_{\succeq}, \tag{3.16}$$

where $\mathbf{z}_{\mathcal{U}}(\mathbf{x}) := \{\mathbf{z}(\mathbf{x}, \boldsymbol{\xi}) \mid \boldsymbol{\xi} \in \mathcal{U}\}$, and $\{\mathbf{z} \in \mathbb{R}^p \mid \mathbf{z} \succeq 0\}$ is denoted by $\mathbb{R}^p_{\succeq}$.

[3] Note that sign $\cdot$ signifies that any vector $\boldsymbol{\xi} \in \mathcal{U}$ can be used as the corresponding function is not uncertain

In Figure 3.4, an example is presented with $\mathcal{U} = \{\boldsymbol{\xi}^1, \boldsymbol{\xi}^2\}$, and $X = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3\}$ to illustrate set-based minmax RE solutions. The boundaries of the respective sets $\{\mathbf{z}_{\mathcal{U}}(\mathbf{x}^j)\}_{j\in\{1,2,3\}}$ are shown using different line-styles. It is evident that $\mathbf{x}^1$ is set-based minmax RE, because $\mathbf{z}_{\mathcal{U}}(\mathbf{x}^1) - \mathbb{R}^2_{\geq}$ does not contain either of the sets $\mathbf{z}_{\mathcal{U}}(\mathbf{x}^2)$ and $\mathbf{z}_{\mathcal{U}}(\mathbf{x}^3)$. Similarly, $\mathbf{x}^2$ is also a set-based minmax RE solution. However, the same could not be said about $\mathbf{x}^3$ since $\mathbf{z}_{\mathcal{U}}(\mathbf{x}^2) \subset \mathbf{z}_{\mathcal{U}}(\mathbf{x}^3) - \mathbb{R}^p_{\geq}$. There are other set-based RE concepts such as *hull-based minmax* RE solutions (Bokrantz and Fredriksson [2017]), *point-based minmax* RE solutions (see Kuroiwa and Lee [2012]), *lower set less ordered efficient, and alternative set less ordered efficient* (see, Ide and Köbis [2014]). In Ide and Köbis [2014], relationships between several RE concepts are investigated, and various special cases are also presented where equivalence is established between a few RE concepts.

The concept of light robustness from Fischetti and Monaci [2009] for SO-RO is generalized for uncertain MOOP in Ide and Schöbel [2016]. The pre-requisite to finding light robust solutions is the existence of a nominal (most likely) scenario. It is quite common to consider a nominal scenario, and it has appeared in many articles such as Ben-Tal and Nemirovski [2002], Ben-Tal et al. [2009]. The motivation behind light robustness is to avoid the overconservativeness of the minmax solutions. For problems with uncertain objective functions and deterministic constraints, the concept of light robustness is defined as follows

Definition 3.9 (Light robustness for SO-RO problems Schöbel [2014]). *Consider a single-objective robust optimization problem $\mathcal{P}(\mathcal{U})$, with $p = 1$, and assume that $\hat{\mathbf{x}} \in X$ is an optimal solution to the problem $\mathcal{P}(\hat{\boldsymbol{\xi}})$, where $\hat{\boldsymbol{\xi}}$ is a nominal scenario. Then, a solution $\mathbf{x} \in X$ is called* lightly robust optimal *to $\mathcal{P}(\mathcal{U})$, w.r.t. $\epsilon \geq 0$ if it is an optimal solution to the min-max problem*

$$\min_{\mathbf{x}\in X} \big\{ \max_{\boldsymbol{\xi}\in\mathcal{U}} z(\mathbf{x}, \boldsymbol{\xi}) \,|\, z(\mathbf{x}, \hat{\boldsymbol{\xi}}) \leq\ z(\hat{\mathbf{x}}, \hat{\xi}) + \epsilon \big\}. \tag{3.17}$$

Ide and Schöbel [2016] has generalized light robustness for multi-objective robust optimization problems with $p > 1$ and $|\mathcal{U}| > 1$ as follows.

Definition 3.10 (Light robustness for robust MOOP). *Given a robust MOOP $\mathcal{P}(\mathcal{U})$, with $p > 1$, and $|\mathcal{U}| > 1$, a nominal scenario $\hat{\boldsymbol{\xi}} \in \mathcal{U}$, and an $\boldsymbol{\epsilon} \in \mathbb{R}^p_{\geq}$, a solution $\mathbf{x}^* \in X$ is called* ϵ-lightly robust efficient solution *for $\mathcal{P}(\mathcal{U})$ if it is one of the efficient solutions to the following deterministic MOOP for a given $\hat{\mathbf{x}} \in X_{\text{eff}}(\hat{\boldsymbol{\xi}})$ (i.e. $\hat{\mathbf{x}}$ is an efficient solution in the nominal scenario)*

$$\min_{\mathbf{x}\in X}\ \big\{ \max_{\boldsymbol{\xi}\in\mathcal{U}}\ \mathbf{z}(\mathbf{x}, \boldsymbol{\xi}) \,|\, z_k(\mathbf{x}, \hat{\boldsymbol{\xi}}) \leq\ z_k(\hat{\mathbf{x}}, \hat{\boldsymbol{\xi}}) + \epsilon_k, \quad k \in \{1, \ldots, p\} \big\}. \tag{3.18}$$

The union of the set of efficient solutions to (3.18) for each $\hat{\mathbf{x}} \in X_{\text{eff}}(\hat{\boldsymbol{\xi}})$ is called ϵ-lightly RE solution set. The main idea behind light robustness for robust MOOP is to find solutions that are good enough in the nominal scenario and to choose the most robust solutions among them. Authors in [Kuhn et al., 2016, Section 4.7] introduced a new RE concept for bi-objective robust optimization problems with uncertain objective function and deterministic constraints. It is called ϵ-representative lightly RE solutions and is aimed to reduce the number of ϵ-lightly robust efficient solutions to be assessed by the decision maker.

For SO-RO problems, it is a common approach to be indifferent toward non-worst-case scenarios. This is sometimes referred to as the so-called *strict robustness*. Iancu and Trichakis [2014], formally prove that the traditional concept of strict robustness (as presented in Ben-Tal et al. [2009]) for SO-RO, which solely focuses on worst-case scenarios, is not reasonable. The main reason is that there might exist alternative solutions that perform much better in other scenarios. Hence, just using the worst-case scenario leaves solutions/decisions un-optimized for other scenarios, which in most problems might be more likely to occur. Hence, (Iancu and Trichakis [2014]) introduced the concept of *Pareto robust optimal* (PRO) solutions for SO-RO problems. To extend the concept of PRO in SO-RO to bi-objective robust optimization problems, [Kuhn et al., 2016, Definition 9] suggest PRO robust efficient (PRO RE) solutions. Hence, it is established that each of the RE solutions must be PRO RE to be non-dominated in all of the scenarios.

Definition 3.11 (Pareto robust optimal (PRO) solutions for SO-RO problems)**.** *Let $\mathcal{U}$ be a set of scenarios, and $z : X \times \mathcal{U} \to \mathbb{R}$ the objective function. Then, a family of functions over the set $\mathcal{U}$ is defined as $\phi_{\mathcal{U}}(\mathbf{x}) := (z(\mathbf{x}, \boldsymbol{\xi}))^{\top}_{\boldsymbol{\xi} \in \mathcal{U}}$ where $\mathbf{x} \in X$, and the function $z(\cdot, \boldsymbol{\xi}) : X \to \mathbb{R}$. A solution $\mathbf{x} \in X$ is PRO if $\nexists \mathbf{x}' \in X$ such that $\phi_{\mathcal{U}}(\mathbf{x}') \preceq \phi_{\mathcal{U}}(\mathbf{x})$.*

For the multi-objective case, an analogous definition is proposed in [Kuhn et al., 2016, Section 5], and referred to as PRO robust efficient (PRO RE) solutions.

Definition 3.12 (PRO RE solutions for robust MOOP)**.** *Let $\mathcal{U}$ be a set of scenarios and $\mathbf{z} : X \times \mathcal{U} \to \mathbb{R}^p$ a p-dimensional vector-valued objective function. Then, a family of vector-valued functions over the set $\mathcal{U}$ is defined as $\phi_{\mathcal{U}}(\mathbf{x}) := (z_1(\mathbf{x}, \boldsymbol{\xi}), \ldots, z_p(\mathbf{x}, \boldsymbol{\xi}))^{\top}_{\boldsymbol{\xi} \in \mathcal{U}}$ where $\mathbf{x} \in X$, and $z_k(\cdot, \boldsymbol{\xi}) : X \to \mathbb{R}$, $k \in \{1, \ldots, p\}$. A solution $\mathbf{x} \in X$ is PRO RE if $\nexists \mathbf{x}' \in X$ such that $\phi_{\mathcal{U}}(\mathbf{x}') \preceq \phi_{\mathcal{U}}(\mathbf{x})$.*

Hence, as discussed in Kuhn et al. [2016], all the RE solutions obtained for a robust MOOP must be PRO RE. This provides a guarantee that no other solution exists that, apart from mitigating the worst-case, also performs better in all other possible scenarios in $\mathcal{U}$.

3.5 Representative set of non-dominated points

Although it is useful to obtain many or all non-dominated points (NDPs), their generation is computationally demanding. Besides, it is then cumbersome for the DM to scan through all the NDPs to identify the most suitable one. There are three types of methods that address such issues *Interactive*, *Inexact* (Metaheuristic), and *Representation* methods. The interactive method requires the active participation of the DM, inexact methods are fast but are not guaranteed to find NDPs. Representation methods identify a subset of NDPs that provide a guarantee for a given performance criterion. There are three different types of performance criteria utilized in representation methods. These are *coverage gap, uniformity* and *cardinality of the representative set*. Firstly, we present a definition of the coverage gap [Ceyhan et al., 2019, Def. 6].

An adaptation to a minimization problem is as follows.

Definition 3.13 (Coverage gap and representative points). *The coverage gap of R with respect to $\mathbf{z} \in Z_{\mathrm{ndp}}$[4] is defined as $\alpha_R(\mathbf{z}) := \min_{\mathbf{y}\in R} \{\max_{i=1,2,3}\{y_i - z_i\}\}$, where $\alpha_R(\mathbf{z}) \geq 0$ holds for $\mathbf{z} \in Z_{\mathrm{ndp}}$. The coverage gap of R with respect to Z_{ndp} is then defined as $\alpha^*_R := \alpha_R(\mathbf{z}^*)$, where $\mathbf{z}^* \in \arg\max_{\mathbf{z}\in Z_{\mathrm{ndp}}} \{\alpha_R(\mathbf{z})\}$. A point $\mathbf{r} \in R$ is said to be representative for the point $\mathbf{z} \in Z_{\mathrm{ndp}}$ if it holds that $\max\limits_{i=1,2,3} \{r_i - z_i\} = \alpha_R(\mathbf{z})$.* □

Solving a representation problem for a given *desired coverage gap* value $\overline{\alpha}$ means identifying a representative set R of Z_{ndp} (set of all the NDPs) such that $\alpha^*_R := \alpha_R(\mathbf{z}^*) \leq \overline{\alpha}$. Note that a reduction of the desired coverage gap $\overline{\alpha}$ may increase the number of points in R and hence, increase the required computing time.

In Sayin [2000], authors presented another performance criterion, so-called, *uniformity*. This relies on evaluating Chebyshev distance between any two points in the representation R of the set Z_{ndp} i.e.,

$$\Delta_R := \min_{\mathbf{y},\mathbf{y}'\in R:\mathbf{y}\neq\mathbf{y}'} \left\{ \max_{i=1,2,3} |y_i - y'_i| \right\}. \tag{3.19}$$

Along with the cardinality of the set R, the three performance criteria form the so-called discrete representation problem which is itself a tri-objective optimization problem as shown in Shao and Ehrgott [2016] and presented as

$$\min_{|R|\geq 2} \{\, \alpha^*_R,\ -\Delta_R,\ |R| \,\}. \tag{3.20}$$

[4] set of all the NDPs

The focus is, however, on the coverage gap in Paper IV due to ease of understanding for the end-users.

4 Summary of included papers

In this chapter, the contributions of the four appended papers are described.

Paper I: Bi-objective optimization of the tactical allocation of job types to machines

In this paper, a bi-objective tactical resource allocation problem (TRAP) is presented. The constraints of this bi-objective MILP are defined in (2.2), and the two objective functions are defined in (2.1). The model is a MILP as the variables $\mathbf{n}$ are continuous, the variables $\mathbf{s}, \mathbf{z}$ are binary, and $\mathbf{x}$ is integral. The makespan minimization of the unrelated parallel machine scheduling problem, i.e. $R||C_{\max}$ is polynomially reducible to the TRAP as shown in Proposition 1 of Paper I. The difficulty in solving the optimization problem stems from the linking constraints (2.2e) which connect the time periods. We propose a starting heuristic (see Section 4 of Paper I) which is based on decomposing the TRAP w.r.t. the time periods, and solving one (smaller) MILP for each time period. The starting heuristic also makes use of Proposition 2 in Paper I (see Proposition 4.1 below), which concludes that for fixed values of $\mathbf{s}$, the variables $\mathbf{z}$ can be regarded continuous. For fixed values of the binary variables $\mathbf{s}$, the following polyhedron is defined in the space of the $\mathbf{z}$ variables:

$$Z(\mathbf{s}, \gamma) := \left\{ \mathbf{z} \,\middle|\, z_{jkt} \in [0,1],\ k \in \mathcal{N}_j,\ j \in \mathcal{J},\ t \in \mathcal{T};\ \text{(2.2e)–(2.2f) hold} \right\}. \quad (4.1)$$

Proposition 2 in Paper I is stated as follows

Proposition 4.1 (On the integrality of the variables $\mathbf{z}$). *For any $s_{jkt} \in \{0,1\}$, $k \in \mathcal{N}_j$, $j \in \mathcal{J}$, $t \in \mathcal{T}$, all the extreme points of the polyhedron $Z(\mathbf{s}, \gamma)$, defined in* (4.1)*, are integral.*

Details of the heuristic are mentioned in Alg. 1 in Paper I. The other main

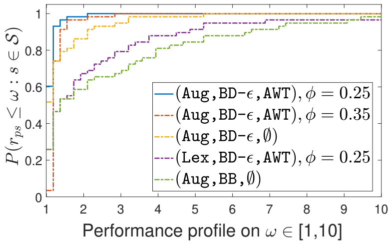


Figure 4.1: Performance profiles of different solution methods while identifying yet-unknown non-dominated points in the entire search area

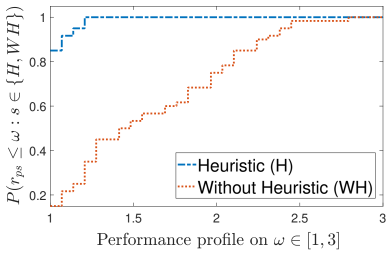


Figure 4.2: Performance profiles when a starting solution is provided (denoted as H), and WH, when no solution is provided

contribution of this work is a modified version of the *bi-directional ϵ-constraint method*. The proposed solution approach combines two well-known criterion space search methods, AWT with adaptive formulae (see Dächert et al. [2012]), and bi-directional ϵ-constraint method (see [Boland et al., 2015, Section 5.1]). It has shown significant positive computational effects on the 60 numerical industrial test cases investigated.

The proposed modification is based on switching to the AWT method when only a pre-defined fraction ϕ of the total search area (in the criterion space) is left to be explored for yet-unknown non-dominated points. In Figure 4.1, solution times of various state-of-the-art solution approaches is compared using the so-called *performance profiles* (see, Dolan and Moré [2002] for details). In Figures 4.1 and 4.2, the term r_{ps} refers to the performance ratio $r_{ps} := \frac{t_{ps}}{\min_{r \in \mathcal{S}}\{t_{pr}\}}$, where $s \in \mathcal{S}$ ($\mathcal{S}$ being the set of solution methods used), and $p \in \mathcal{P}$ ($\mathcal{P}$ being the set of problem instances), and t_{ps} is the computing time used for solving problem instance p by solution method s.

For each solution method in Figure 4.1, the two objectives are tackled by either augmentation (Aug) or a lexicographic (Lex) minimization of the two. The main criterion space search methods used are the bi-directional ϵ-constraint (BD-ϵ) and the balanced box (BB) method. A switch to the AWT method is denoted by AWT, while $\emptyset$ means that there is no such switch. The solution approaches compared are defined by the 3-tuples (Aug,BD-ϵ,AWT), (Aug,BD-ϵ,$\emptyset$), (Aug,BB,$\emptyset$), and (Lex,BD-ϵ,AWT), including two variants of (Aug,BD-ϵ,AWT) for the values $\phi \in \{0.25, 0.35\}$. Hence, in total five variants are tested and presented in Figure 4.1. In Figure 4.2, the performance profiles illustrate the effect of using a starting feasible solution.

Paper II: Robust optimization of a bi-objective tactical resource allocation problem with uncertain qualification costs

In this paper, the qualification cost parameters $\boldsymbol{\beta}_{jk}, j \in \mathcal{J}$ and $k \in \mathcal{N}_j$, the coefficients of the second objective function (g_2, i.e. (2.1b)) are considered uncertain. In (2.4) an uncertain bi-objective optimization problem is presented, with an uncertain objective function and possessing the same feasible set (2.2) as in the deterministic TRAP. We have two main contributions. Firstly, we have presented a new robust efficiency concept called *positive robustness ϵ-representative lightly RE solution*. This new RE concept is presented as an alternative to ϵ-representative lightly RE solution, as the former captures the positive effect on mitigating risk by replacing an efficient solution in the nominal scenario with a solution that is "good enough" in the nominal scenario and has a net reduction in qualification cost in the worst-case scenario.

The second contribution is a new solution approach called 3-*stage approach*, involving the solution of two bi-objective optimization problems to find all the desired PRO RE solutions instead of solving a tri-objective optimization problem as suggested in Kuhn et al. [2016] for bi-objective optimization problems with one uncertain objective function and two scenarios. Our proposed approach is computationally superior than the one presented in Kuhn et al. [2016]. We use the quadrant shrinking method (QSM) for the purpose of solving the tri-objective IP, hence the name QSM in Figure 4.3. For almost all the instances (except instance 11) our proposed approach finds a smaller number of PRO RE solutions but manages to find all the ones that are interesting for the decision-makers to analyze. However, as it is evident from Fig. 4.3, the solution times of our 3-stage approach are significantly lower than those of the QSM method.

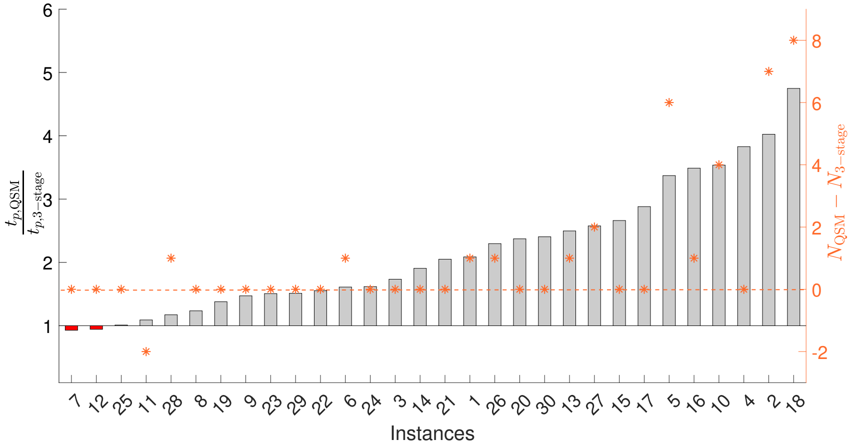
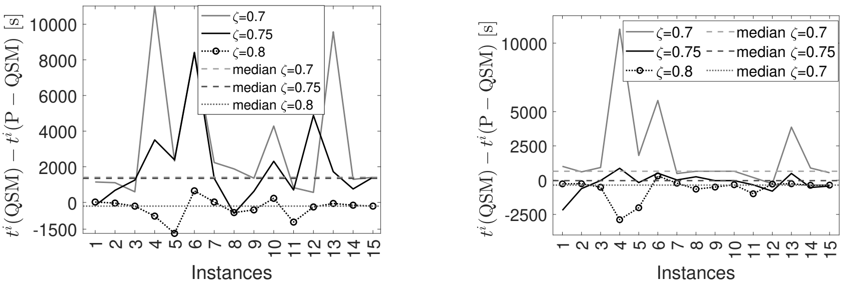


Figure 4.3: Left axis: Ratio of solutions times (grey bars; red bars for negative values). Right axis: Difference between the number of PRO-RE solutions identified by the QSM (N_{QSM}) and the 3-stage method ($N_{3\text{-stage}}$) (orange asterisk). All computed sets are minimal sets

Paper III: A criterion space decomposition approach to generalized tri-objective tactical resource allocation

In Paper III, we have introduced inventory in the TRAP. We call this new model the Generalized TRAP (GTRAP), which is described in detail in Sec. 3.1 in Paper III. This allows the model to have an inventory of even semi-finished parts that can have a significant effect on the previously defined two objectives as well. Furthermore, we have also developed a specialized procedure that relies on partitioning the criterion space and solving sub-problems simultaneously to reduce computational time. The main idea is to split the projected two-dimensional criterion space using Prop. 9 and 10 from Paper III and use the QSM Boland et al. [2017] to solve each sub-problem. Furthermore, some modifications are suggested to avoid some of the redundancies due to parallelization in Sec. 4.3.1 in Paper III. In Fig. 4.4b and 4.4a we illustrate the difference in solution time of using the QSM (t^i(QSM)) and our proposed parallel QSM (t^i(P-QSM)) in seconds, where i is the problem instance. We have also performed sensitivity analyses for different values of threshold $\zeta_k = \zeta \in \{0.7, 0.75, 0.80\}$, $k \in \mathcal{K}$.



(a) $\delta_1 = 0.05$ **(b)** $\delta_1 = 0.1$

Figure 4.4: Difference of solution time for QSM and P-QSM [s] for threshold values $\zeta \in \{0.70, 0.75, 0.80\}$ and varied the discretization $\delta_1 \in \{0.05, 0.1\}$ used for the first objective function which is not integer-valued

Paper IV: A method to identify a representative set of non-dominated points for discrete tri-objective optimization problems

During the course of writing Paper I–III, we realized that for a significant reduction in solution time, we need to reduce the number of NDPs identified. However, we do not want to use in-exact methods as there will be no guarantee on any of the three performance criteria mentioned in Sec. 3.5. There exist several interesting works that are discussed in detail in Sec. 1.2 in Paper IV. Our contribution to this paper is developing a new method specialized for any discrete tri-objective optimization problem. The method utilizes some aspects of the approach presented in Paper III and is generalized for any discrete tri-objective optimization problem. Furthermore, new modifications are introduced that were not considered in the P-QSM from Paper III.

The computational performance of our proposed algorithm is far superior compared to the state-of-the-art methods on industrial instances of GTRAP. Furthermore, we also tested our approach for some other general instances such as the multi-dimensional three-objective knapsack problem. We compare our algorithm $\overline{\alpha}$-PQSM with two state-of-the-art algorithms grid-point-based algorithm GPBA-A Mesquita-Cunha et al. [2023] and the territory-defining algorithm (TDA) Ceyhan et al. [2019]. The results are also promising for these instances. A performance profile is illustrated in Fig. 4.5 for the multi-dimensional knapsack problem instances. The performance metric is the ratio of solution time similar to the one described in the summary of Paper I. Here, we have three algorithms in the set $\mathcal{S}$ and 300 problem instances in $\mathcal{P}$.

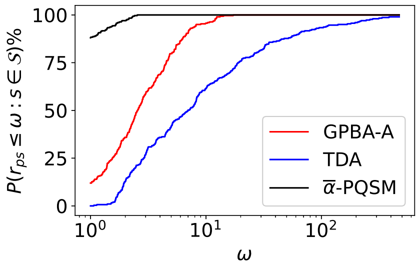


Figure 4.5: Performance profile for different algorithms using the ratio of solution time as the performance measure for multi-dimensional three-objective knapsack instances.

5 Conclusion and future research areas

Conclusion. In the Tactical Resource Allocation Problem, the primary objective is to balance machine loading levels in order to reduce product lead times. In Papers I and II, this balance is achieved by introducing qualifications, which entail one-time setup costs. Furthermore, Paper III incorporates inventories and assesses their impact on the other two objectives.

From a practical standpoint, certain categories of machines consistently display high loading levels. For example, multi-task machines capable of performing various operations are typically highly loaded and often become bottlenecks. However, by adding new qualifications, these loading levels can be mitigated. In Paper II, we emphasized the need for a careful assessment of variations in qualification costs, as this can considerably affect the trade-off between loading levels and qualification costs in certain scenarios. In some instances, alternative efficient solutions may be identified. In Paper III, we employed a series-assembly structure for the final products and allowed for the accumulation of inventories of both semi-finished and finished products. This strategy enabled us to uncover new solutions that were not feasible in Papers I and II, although at the expense of additional inventory, which is accounted for as a third objective. Our study underscores the potential benefits of maintaining certain inventories, which, in specific cases, may reduce the need for expensive qualifications. Lastly, in Paper IV, we proposed an innovative algorithm aimed at identifying a representative set of non-dominated points, thereby alleviating the computational load on decision-makers and enhancing the tool's usability.

Future research areas. Some of the future research areas envisioned are

Stochastic Tactical resource allocation model: We believe that there is a need to develop a new framework regarding the stochastic arrival of raw materials. This extension should be done to the GTRAP presented in Paper III. However, this requires a deep dive into understanding how the delays in the delivery of raw materials are distributed and their dependencies on properties which depend on the statuses of suppliers. We have an ongoing manuscript (not considered in this thesis for evaluation) in which we develop a model that takes into account the stochastic arrival of raw materials. For this purpose, we need to define smaller time buckets in order to capture the effect of the late arrival of raw materials on the excess loading, qualification costs, and inventory. We then model the uncertainties using a truncated geometric distribution.

Visualization and user-interface: There is a possibility to explore preference articulation. This can involve interactive methods, visual analytics, and preference modeling techniques such as Inverse optimization. More effort is needed in learning from real production data to find better ways to combine the objectives.

Recommender system: I proposed and supervised a master thesis project with GKN Aerospace entitled *Evaluating compatibility between machines and operations for aerospace engine products* Costanzo and Limbayyaswamimath [2021] which was aimed as a starting point to develop a machine learning system for recommending job types to machines based on historical data. However, the lack of unified classification and production-level data led to no further progress but it is definitely a good idea for future research as data collection is increasing.

Computational studies: The frameworks developed regarding robust optimization, criterion space decomposition, and representative set of non-dominated points could be adapted and applied to other discrete bi- and tri-objective optimization problems, including computational studies of specific and realistic instances of these.

Bibliography

Y. P. Aneja and K. P. K. Nair. Bicriteria transportation problem. *Management Science*, 25(1):73–78, 1979. doi: b2t226.

Aharon Ben-Tal and Arkadi Nemirovski. Robust optimization: methodology and applications. *Mathematical Programming*, 92(3):453–480, 2002. doi: dw69js.

Aharon Ben-Tal, L.E. Ghaoui, and A. Nemirovski. *Robust Optimization.* Princeton University Press, 2009.

H. P. Benson. Existence of efficient solutions for vector maximization problems. *Journal of Optimization Theory and Applications*, 26(4):569–580, 1978. doi: ffwsmd.

Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004. doi: d26x3s.

E. Robert Bixby, Mary Fenelon, Zonghao Gu, Ed Rothberg, and Roland Wunderling. MIP: Theory and practice—closing the gap. In *System Modelling and Optimization*, pages 19–49, Boston, MA, 2000. Springer US.

John H. Blackstone Jr. *APICS Dictionary* 14^{th} *edition*. Association for Supply Chain Management, 2013.

Rasmus Bokrantz and Albin Fredriksson. Necessary and sufficient conditions for Pareto efficiency in robust multiobjective optimization. *European Journal of Operational Research*, 262(2):682–692, 2017. doi: gbn4tt.

Natashia Boland, Hadi Charkhgard, and Martin Savelsbergh. A criterion space search algorithm for biobjective integer programming: The balanced box method. *INFORMS Journal on Computing*, 27(4):735–754, 2015. doi: f72s8v.

Natashia Boland, Hadi Charkhgard, and Martin Savelsbergh. The Quadrant Shrinking Method: A simple and efficient algorithm for solving tri-objective integer programs. *European Journal of Operational Research*, 260(3):873–885, 2017. doi: ftpq.

V. Joseph Bowman. On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives. In *Multiple Criteria Decision Making*, pages 76–86, Berlin, Heidelberg, 1976. Springer Berlin Heidelberg. doi: gcwh.

James R. Bradley and Peter W. Glynn. Managing capacity and inventory jointly in manufacturing systems. *Management Science*, 48(2):273–288, 2002. doi: bm3pr5.

Gökhan Ceyhan, Murat Köksalan, and Banu Lokman. Finding a representative nondominated set for multi-objective mixed integer programs. *European Journal of Operational Research*, 272(1):61–77, 2019. doi: jccf.

L. G. Chalmet, L. Lemonidis, and D. J. Elzinga. An algorithm for the bi-criterion integer programming problem. *European Journal of Operational Research*, 25 (2):292–300, 1986. doi: frc52z.

V. Chankong and Y. Y. Haimes. *Multiobjective Decision Making: Theory and Methodology*. Elsevier Sceince New York, 1983.

Shih-Pin Chen and Wen-Lung Huang. A membership function approach for aggregate production planning problems in fuzzy environments. *International Journal of Production Research*, 48(23):7003–7023, 2010. doi: chbpvb.

Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer and Combinatorial Optimization*. Springer International Publishing, 2014.

Gérard Cornuéjols. Revival of the Gomory cuts in the 1990's. *Annals of Operations Research*, 149(1):63–66, 2006. doi: ftqzfx.

Roberto Costanzo and Vedanth Limbayyaswamimath. Evaluating compatibility between machines and operations for aerospace engine products. Master's thesis, Chalmers University of Technology, Sweden, 2021. URL `https://hdl.handle.net/20.500.12380/302421`.

Harlan Crowder, Ellis L. Johnson, and Manfred Padberg. Solving large-scale zero-one linear programming problems. *Operations Research*, 31(5):803–834, 1983. doi: ddskh8.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi: bnw2vv.

Manuel Díaz-Madroñero, Josefa Mula, and David Peidro. A review of discrete-time optimization models for tactical production planning. *International Journal of Production Research*, 52(17):5171–5205, 2014. doi: ftpt.

Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002. doi: cpzm5c.

Kerstin Dächert, Jochen Gorski, and Kathrin Klamroth. An augmented weighted Tchebycheff method with adaptively chosen parameters for discrete bicriteria optimization problems. *Computers & Operations Research*, 39 (12):2929–2943, 2012. doi: f4kd8m.

Kerstin Dächert, Kathrin Klamroth, Renaud Lacour, and Daniel Vanderpooten. Efficient computation of the search region in multi-objective optimization. *European Journal of Operational Research*, 260(3):841–855, 2017. doi: gbg9.

Matthias Ehrgott. *Multicriteria Optimization*. Springer-Verlag, 2005. doi: cgtkmr.

Matthias Ehrgott. A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research*, 147(1):343–360, 2006. doi: bhpt3n.

Matthias Ehrgott, Jonas Ide, and Anita Schöbel. Minmax robustness for multi-objective optimization problems. *European Journal of Operational Research*, 239 (1):17–31, 2014. doi: ftpm.

Gabriele Eichfelder and Leo Warnow. A hybrid patch decomposition approach to compute an enclosure for multi-objective mixed-integer convex optimization problems. *Mathematical Methods of Operations Research*, aug 2023. doi: kprq.

Matteo Fischetti and Michele Monaci. *Light Robustness*, pages 61–84. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-05465-5. doi: cz8zj4.

S. Fotedar, A.-B. Strömberg, E. Åblad, and T. Almgren. Robust optimization of a bi-objective tactical resource allocation problem with uncertain qualification costs. *Autonomous Agents and Multi-Agent Systems*, 36:1–31, 2022. doi: gqd9k6.

S. Fotedar, A.-B. Strömberg, and T. Almgren. Bi-objective optimization of the tactical allocation of job types to machines: mathematical modeling, theoretical analysis, and numerical tests. *International Transactions in Operational Research*, 30(6):3479–3507, 2023a. doi: gqmrww.

Sunney Fotedar, Ann-Brith Strömberg, Torgny Almgren, and Stefan Cedergren. A criterion space decomposition approach to generalized tri-objective tactical resource allocation. *Computational Management Science*, 20(1), 2023b. doi: j4hv.

Michael R. Garey and David S. Johnson. *A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

P. Genin, S. Lamouri, and A. Thomas. Multi-facilities tactical planning robustness with experimental design. *Production Planning & Control*, 19(2):171–182, 2008. doi: dgfdxd.

Marc Goerigk and Anita Schöbel. *Algorithm Engineering in Robust Optimization*, pages 245–279. Springer International Publishing, Cham, 2016. doi: gbhc.

Zonghao Gu, George L. Nemhauser, and Martin W. P. Savelsbergh. Lifted flow cover inequalities for mixed 0-1 integer programs. *Mathematical Programming*, 85:439–467, 1999. doi: b433vd.

Anshuman Gupta and Costas D. Maranas. A hierarchical Lagrangean relaxation procedure for solving midterm planning problems. *Industrial & Engineering Chemistry Research*, 38(5):1937–1947, 1999. doi: dwhm4r.

Walter J. Gutjahr and Alois Pichler. Stochastic multi-objective optimization: a survey on non-scalarizing methods. *Annals of Operations Research*, 236(2): 475–499, 2013. doi: gbhb.

Karla L. Hoffman and Manfred Padberg. Improving LP-representations of zero-one linear programs for branch-and-cut. *ORSA Journal on Computing*, 3 (2):121–134, 1991. doi: dhm77p.

Dan A. Iancu and Nikolaos Trichakis. Pareto efficiency in robust optimization. *Management Science*, 60(1):130–147, 2014. doi: ftpf.

Jonas Ide and Elisabeth Köbis. Concepts of efficiency for uncertain multi-objective optimization problems based on set order relations. *Mathematical Methods of Operations Research*, 80(1):99–127, 2014. doi: f6df4g.

Jonas Ide and Anita Schöbel. Robustness for uncertain multi-objective optimization: a survey and analysis of different concepts. *OR Spectrum*, 38(1): 235–271, 2016. doi: ftpj.

Gokhan Kirlik and Serpil Sayın. A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *European Journal of Operational Research*, 232(3):479–488, 2014. doi: ftpr.

Panos Kouvelis and Gang Yu. *Approaches for Handling Uncertainty in Decision Making*, pages 333–356. Springer US, Boston, MA, 1997. doi: gbgz.

K. Kuhn, A. Raith, M. Schmidt, and A. Schöbel. Bi-objective robust optimisation. *European Journal of Operational Research*, 252(2):418–431, 2016. doi: ftpc.

Daishi Kuroiwa and Gue Myung Lee. On robust multiobjective optimization. *Vietnam Journal of Mathematics*, 40(2 & 3):305–317, 2012.

Adam J. Lewestam and Henrik S.U. Mäki. Increasing throughput potential in functional oriented machining plants. Master's thesis, Chalmers University of Technology, Sweden, 2015. URL `https://odr.chalmers.se/handle/20.500.12380/219088`.

Banu Lokman and Murat Köksalan. Finding all nondominated points of multi-objective integer programs. *Journal of Global Optimization*, 57(2):347–365, 2012. doi: f5crcw.

Mariana Mesquita-Cunha, José Rui Figueira, and Ana Paula Barbosa-Póvoa. New ε-constraint methods for multi-objective integer linear programming: A Pareto front representation approach. *European Journal of Operational Research*, 306(1):286–307, 2023. doi: jccb.

Kaisa Miettinen. *Nonlinear Multiobjective Optimization*. Springer Science Business Media New York, 1988.

Hokey Min and Gengui Zhou. Supply chain modeling: past, present and future. *Computers & Industrial Engineering*, 43(1–2):231–249, 2002. doi: btm8vj.

Andreas Myrelid and Jan Olhager. Hybrid manufacturing accounting in mixed process environments: A methodology and a case study. *International Journal of Production Economics*, 210:137–144, 2019. doi: gjcx78.

Julia Niebling and Gabriele Eichfelder. A branch–and–bound-based algorithm for nonconvex multiobjective optimization. *SIAM Journal on Optimization*, 29 (1):794–821, jan 2019. doi: kprr.

Yves Pochet and Laurence A. Wolsey. *Production Planning by Mixed Integer Programming*. Springer New York, 2006. doi: fjz74x.

Anthony Przybylski, Xavier Gandibleux, and Matthias Ehrgott. Two phase algorithms for the bi-objective assignment problem. *European Journal of Operational Research*, 185(2):509–533, 2008. doi: bx5kz2.

Serpil Sayin. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming, Series B*, 87(3):543–560, 2000. doi: b3jvth.

Anita Schöbel. Generalized light robustness and the trade-off between robustness and nominal quality. *Mathematical Methods of Operations Research*, 80(2): 161–191, 2014. doi: f6m2qf.

Lizhen Shao and Matthias Ehrgott. Discrete representation of non-dominated sets in multi-objective linear programming. *European Journal of Operational Research*, 255(3):687–698, 2016. doi: jpxz.

Roman Slowinski and Jacques Teghem, editors. *Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty*. Springer Netherlands, 1990. doi: fq7dp9.

Ralph E. Steuer and Eng-Ung Choo. An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3):326–344, 1983. doi: bv79s7.

Thomas Stidsen, Kim Allan Andersen, and Bernd Dammann. A branch and bound algorithm for a class of biobjective mixed integer programs. *Management Science*, 60(4):1009–1032, 2014. doi: gbg6.

John Sylva and Alejandro Crema. A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Research*, 158(1):46–55, 2004. doi: b5ksfw.

K. Thörnblad, A.-B. Strömberg, M. Patriksson, and T. Almgren. Scheduling optimisation of a real flexible job shop including fixture availability and preventive maintenance. *European Journal of Industrial Engineering*, 9(1):126–145, 2015. doi: f3nx7f.

Sandra Transchel, Stefan Minner, Josef Kallrath, Nils Löhndorf, and Ulrich Eberhard. A hybrid general lot-sizing and scheduling formulation for a production process with a two-stage product structure. *International Journal of Production Research*, 49(9):2463–2480, 2011. doi: czp9tx.

Jan A. Van Mieghem. Capacity management, investment, and hedging: Review and recent developments. *Manufacturing & Service Operations Management*, 5 (4):269–302, 2003. doi: dhx6pz.

Thomas Vincent, Florian Seipp, Stefan Ruzika, Anthony Przybylski, and Xavier Gandibleux. Multiple objective branch and bound for mixed 0-1 linear programming: Corrections and improvements for the biobjective case. *Computers & Operations Research*, 40(1):498–509, 2013. doi: ftpv.

M. Visée, J. Teghem, M. Pirlot, and E. L. Ulungu. Two-phases method and branch and bound procedures to solve the bi–objective knapsack problem. *Journal of Global Optimization*, 12(2):139–155, 1998. doi: dw873c.

Cansheng Wei, Yongjian Li, and Xiaoqiang Cai. Robust optimal policies of production and inventory with uncertain returns and demand. *International Journal of Production Economics*, 134(2):357–367, 2011. doi: bv67qg.

Andrzej P. Wierzbicki, Marek Makowski, and Jaap Wessels, editors. *Model-Based Decision Support Methodology with Environmental Applications*. Springer Netherlands, 2000. doi: gg2g.