Preface

Optimized object packing (OOP) studies are aimed at finding the best possible non-overlapping arrangement of a given set of objects in a container (or a set of containers). This very general modeling paradigm can be specified in great many ways, thereby leading to interesting—and, as a rule, challenging—optimization models. OOPs can be important components, e.g., in cutting, covering, layout design, loading, scheduling, and supply chain management studies. Arguably, OOP is among the most significant application areas of operations research. Let us remark additionally that the study of atomic or molecular conformations, spherical point arrangements, the design of experiments and other related areas in computational physics, chemistry, biology, and numerical analysis are closely related to the OOP subject.

While the depth and quality of the decisions required to find high-quality OOPs is increasing, we have also witnessed significant and continuing progress regarding both theoretical advances and ready-to-use tools for actual OOP applications. Theoretical advances, scientific innovation, and algorithmic development are supported and enhanced by today's state-of-the-art computational modeling and optimization environments. Until quite recently, the numerical optimization approaches to tackle OOPs were essentially limited to handle convex (linear or nonlinear, continuous) optimization problems and linearly structured combinatorial and mixed integer-continuous optimization problems. The consideration of integer decision variables in more flexible nonlinear modeling frameworks gives rise to even harder combinatorial and mixed integer-continuous optimization problems. The solution of such computational challenges is becoming increasingly more viable.

In addition to the long-time theoretical interest directed towards OOPs, there is a strong practical motivation to solve various real-world packing problems. Our aim has been to offer a selection of efficient exact and heuristic algorithmic approaches and practical case studies related to the broadly interpreted subject of OOP. The contributing authors are well-recognized researchers and practitioners working (also) in the area of OOP-related modeling and optimization. Next we provide an overview of the contributed chapters (ordered on the basis of the family name of their first authors).

Chapter 1, titled "Using a Bin Packing Approach for Stowing Hazardous Containers into Containerships," has been authored by Daniela Ambrosino and Anna Sciomachen. They address the problem of determining stowage plans for containers loaded into a ship. This is the so-called master bay plan problem (MBPP). The MBPP consists in determining how to stow a set of containers—split into groups according to their size, type, class of weight, and destination—into a set of available slots (locations either on the deck or in the hold area) of predetermined bays of a container ship. Context-dependent structural and operational constraints, related both to the containers and to the ship, have to be satisfied by the MBPP. As an important variant of the MBPP, in this chapter the stowage of hazardous containers is considered. The need for stowing dangerous goods implies additional constraints concerning the safety of the entire cargo, since dangerous goods (categorized into different types) have to be stowed away from certain other goods. This variant of the MBPP is handled on the basis of its relationship with the bin packing problem, in which the packed items are containers and the bins are sections of the ship available for the stowage of hazardous (as well as other) containers. Following a step-by-step procedure for properly loading all containers on board, Ambrosino and Sciomachen show how the segregation rules derived from the International Maritime Dangerous Goods Code affect the available slots of the bins. The chapter reports a real-life case study solved by using the commercial software package CPLEX.

Chapter 2, titled "Dynamic Packing with Side Constraints for Datacenter Resource Management," has been written by Sophie Demassey, Fabien Hermenier, and Vincent Kherbache. Datacenter Resource Management (DRM) requires the assignment of virtual machines (VMs) with dynamically changing resource demands to physical machines with dynamically changing available capacities. The changes occurring at runtime invalidate the currently given assignments, thereby necessitating their updates (adjustments). The assignments are also subject to changing restrictions that express various datacenter user requirements. Within this context, the chapter surveys the application of vector packing (called the VM reassignment problem) providing insight into its dynamic and heterogeneous nature. The study advocates flexibility to answer the issues highlighted above, and presents BtrPlace, an open source resource manager based on the discipline of Constraint Programming. BtrPlace offers a flexible and scalable solution procedure as illustrated by sizeable numerical examples. The authors' experiments show that BtrPlace can effectively manage thousands of web applications running on thousands of physical machines.

Chapter 3, titled "Packing Optimization of Free-Form Objects in Engineering Design," has been authored by Georges M. Fadel and Margaret M. Wiecek. OOPs arising in the engineering design context—often referred to as layout optimization problems—require the determination of the arrangement of given subsystems or components within some enclosure (area or volume), to achieve a given set of objectives in the presence of spatial and/or performance constraints. As a rule, such optimization problems are challenging, due to their highly multimodal structure. In addition, the problems are often described by models that may not have closed-form analytical representations, and/or may require the use of computationally expensive evaluation procedures. The time needed to resolve object intersection calculations can increase exponentially with the number of objects to be packed, while the space available for the placement of these components becomes increasingly scarce. The chapter reviews the results of a multi-year research effort, specifically targeting the development of computational tools for automotive engineering design. The packing problems discussed are represented by single- or multi-objective optimization problems. The solution approaches reviewed rely on evolutionary algorithms, due to the level of complexity that precludes the use of sufficiently effective exact methods.

Chapter 4, titled “A Modeling-Based Approach for Non-standard Packing Problems,” has been written by Giorgio Fasano. The chapter is focused on packing tetris-like items orthogonally, with the possibility of rotations into a convex domain, in the optional presence of additional constraints. Mixed Integer Linear Programming (MILP) and Mixed Integer Nonlinear Programming (MINLP) model versions, previously studied by the author, are reviewed. An efficient formulation of the objective function, aimed at maximizing the loaded cargo, is given as an MILP model. The MINLP model has been developed to address the relevant feasibility sub-problem: its purpose is to improve approximate solutions, as an intermediate step of a heuristic process. A space-indexed model is also introduced and the problem of approximating polygons by means of tetris-like items is studied. In both cases an MILP formulation has been adopted. Finally, a heuristic approach is proposed to provide effective solutions in practical applications.

Chapter 5, titled “CAST: A Successful Project in Support of the International Space Station Logistics,” has been authored by Giorgio Fasano, Claudia Lavopa, Davide Negri, and Maria Chiara Vola. The International Space Station (ISS) is one of the most challenging currently active space programs: this program requires the handling of demanding logistic issues, mainly in relation to on-orbit maintenance and resource resupply. To serve the ISS, a fleet of launchers and vehicles is made available by the space agencies involved. An overall traffic plan schedules the recurrent upload and download interventions between the Earth and the ISS orbit. The European Space Agency (ESA) contributed annually to the ISS logistics from 2008 to 2014, by accomplishing five Automated Transfer Vehicle (ATV) missions. Within the related cargo accommodation context, in addition to tight balancing conditions, difficult packing issues arose: these had to be solved under conditions of strict deadlines and possible last minute changes. The Cargo Accommodation Support Tool (CAST) is a dedicated optimization framework funded by ESA and developed by Thales Alenia Space to create the ATV cargo accommodation plan. The chapter first describes the ATV loading problem. The basic concept of CAST is then reviewed, highlighting the advantages of the methodology adopted, both in terms of solution quality and time savings. Current extensions and possible future enhancements are also discussed.

Chapter 6, titled “Cutting and Packing Problems with Placement Constraints,” has been written by Andreas Fischer and Guntram Scheithauer. In real-life cutting and packing problems additional placement constraints are often present.

For instance, defective regions of some raw material cannot become part of the end products. More generally, due to varying quality requirements, certain products may contain (material) parts of lower quality, while this is not allowed for some other products. The chapter considers one- and two-dimensional rectangular cutting and packing problems, in which items of given types have to be cut from (or placed on) a given raw material in such a way that optimizes the value of a context-specific objective function. In the one-dimensional (1D) case, it is assumed that for each item type the allocation intervals (segments of the raw material) are given, so that all items of the same type must be contained by one of these allocation intervals. The authors also consider problems in which the length of the 1D items could vary within known tolerances. In the two-dimensional (2D) case, rectangular items of different types have to be cut from a large rectangle. Here the authors investigate guillotine cutting plans under the condition that defective rectangular regions are not allowed to be part of the manufactured products. For these scenarios they present solution strategies which rely on the branch-and-bound principle or on dynamic programming. Based on the properties of the corresponding objective functions, they discuss possibilities to reduce computational complexity. This includes the definition of appropriate sets of potential allocation (cut) points which have to be inspected to obtain an optimal solution. Applying dominance considerations, the set of such allocation points can be kept small. In particular, the computational complexity becomes independent of the unit of measure of the input data. Possible generalizations of the solution strategy are also discussed.

Chapter 7, titled "A Container Loading Problem MILP-Based Heuristics Solved by CPLEX: An Experimental Analysis," has been authored by Stefano Gliozzi, Alessandro Castellazzo, and Giorgio Fasano. They consider a standard container loading model form: placing smaller boxes orthogonally (generally with the possibility of rotations) into a larger box, to maximize the loaded volume. Although this problem is NP-hard, a number of algorithms can handle it with high numerical efficiency. The task becomes even more challenging when additional conditions with an overall impact have to be taken into account. In such cases, a modeling-based global scope approach is advocated, e.g., when considering load balancing requirements. Mixed Integer Linear Programming (MILP) models relevant to the container loading problem including possible extensions are available in the literature. An MILP model, presented in Chap. 4 of this book, is taken as a basis. The chapter discusses some important computational aspects of the container loading problem in its classical form (i.e., without additional conditions). An *ad hoc* heuristics, derived from the above-mentioned overall approach, is also outlined. Next, the use of CPLEX as an MILP optimizer is considered. Case studies concerning the solution of the MILP model *tout court* for smaller model instances are reported first. Outcomes relevant to the *ad hoc* heuristics are shown next, in relation to a number of more difficult instances. Examples of container loading problems, involving additional balancing conditions, are also presented.

Chapter 8, titled "Automatic Design of Optimal LED Street Lights," has been written by Balázs L. Lévai and Balázs Bánhelyi. The authors discuss the issue of light pollution—i.e., the unnecessary lighting of outdoor areas—which has negative consequences, e.g., by disturbing wild life, not to mention energy conservation aspects. Based on its capabilities, light-emitting diode (LED) technology offers an efficient solution to this problem. LEDs have many advantages over incandescent light sources including lower energy consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. Many cities in developed countries have LED street lights. Designing the orientation of LEDs in street lights is a nontrivial problem, however, since the use of multiple LED packages is required to replace a single incandescent light bulb. Specifically, the positional angles of LEDs in lamps have to be determined to produce an even light distribution over the target surface. Determining the set of best angles is a global optimization (GO) problem, induced by the underlying task of target area covering problems. The authors present an automatic design approach to find suitable LED configurations for street lights, including an embedded light pattern computation technique to evaluate these configurations. The resulting GO problems are solved (heuristically) using a genetic algorithm. In order to speed up the design process, a possible way of parallelization focused on the light pattern computation module is also discussed.

Chapter 9, titled "Approximate Packing: Integer Programming Models, Valid Inequalities and Nesting," has been authored by Igor Litvinchev, Luis Infante, and Lucero Ozuna. They suggest the use of a regular grid to approximate the container to be loaded. This way, the object packing problem is reduced to assigning objects to nodes of the grid, subject to non-overlapping constraints. This approximate packing problem is then formulated as a large-scale linear binary optimization problem. Different model formulations to express the non-overlapping constraints are presented and compared, and valid inequalities are proposed to strengthen the formulations. This general approach is applied first to the packing of circular and L-shaped objects into a rectangular container. Circular objects are defined in the general sense, as a set of points that are located at the same (not necessary Euclidean) distance from a given point. Different objects—including ellipses, rhombuses, rectangles, and octagons—can be handled by simply changing the definition of the norm used to define the distance concept. Nesting objects inside one another is also considered when appropriate, in the context of certain applications. Numerical results are presented to demonstrate the efficiency of the proposed approach: the optimization problems are solved using CPLEX.

Chapter 10, titled "Exploiting Packing Components in General-Purpose Integer Programming Solvers," has been written by Jakub Mareček. The author discusses the task of packing boxes into a large box; this task is often only a part of a more complex problem. As an example, in furniture supply chain applications, one needs to decide which trucks to use to transport furniture between production sites and distribution centers or stores: obviously, one has to search for packings that guarantee that all delivery items fit into the available trucks. Such problems are often formulated and solved using general-purpose integer programming solvers. This chapter studies the problem of identifying a compact formulation of the packing component in a general instance of integer linear programming. The space-indexed approach advocated is based on exploiting the problem structure and a reformulation using the adaptive discretization proposed by Allen, Burke, and Mareček, and then solving the extended reformulation. The solvers tested were CPLEX, Gurobi, and SCIP, with CLP as the linear programming solver. Results related to solving model instances with up to 10,000,000 boxes are reported.

Chapter 11, titled “Robust Designs for Circle Coverings of a Square,” has been authored by Mihály Csaba Markót. The chapter investigates coverings of a square by a set of uniform size circles of optimized (minimal) radius, when uncertainties are present regarding the actual locations of the circles. This model statement is related to deploying sensors or other kinds of observation units with possible uncertainties regarding their actual deployments. Application examples include scenarios when the deployment has to be made remotely (e.g., from the air) into a potentially dangerous environment, or into a location with unknown terrain, or it is influenced by the weather conditions. The goal of the study is to produce coverings that are optimal in terms of a minimal radius, and are also robust in the following sense: wherever the circles are actually placed within a given uncertainty region, the end result is still guaranteed to be a covering. Markót investigates three special uncertainty regions: first he proves that for uniform circular uncertainty regions the optimal robust covering can be created from the exact optimal covering without uncertainties, provided that the exact covering configuration is feasible for the robust scenario. For uncertainty regions given by line segments and by general convex polygons, he proposes a bi-level optimization method combining a complete and rigorous global search and a derivative free black-box search. Numerical examples illustrate the efficiency of the suggested approach.

Chapter 12, titled “Batching-Based Approaches for Optimized Packing of Jobs in the Spatial Scheduling Problem,” has been written by Sudharshana Srinivasan, J. Paul Brooks, and Jill Hardin Wilson. Spatial scheduling problems (SSPs) involve the non-overlapping arrangement of jobs within a limited physical workspace in such a manner that some scheduling objective is optimized. In the context of shipbuilding and other large-scale manufacturing industries, the jobs typically occupy large areas, requiring that the same contiguous units of space be assigned throughout the duration of their processing time. This adds an additional level of complexity to the corresponding scheduling problem. Since solving large-scale problem instances by using exact methods becomes computationally intractable, there is a need to develop efficient alternative strategies to provide near-optimal solutions. Much of the literature focuses on minimizing the makespan of the schedule. The authors propose two heuristic methods to minimize the sum of completion times. The approach is based on grouping jobs into batches and then applying a scheduling heuristic to these batches. It is shown that grouping jobs earlier in the schedule can result in poor performance when the jobs have large differences in processing times. The authors provide bounds on the performance of the algorithms, and present computational results comparing the solutions to the optimal objective obtained from the integer programming formulation for SSP. For a smaller number of jobs, both algorithms produce comparable solutions. For instances with a larger number of jobs and a higher variability in spatial dimensions, the efficient area-based model outperforms the iterative model, both in terms of solution quality and run time.

Chapter 13, titled "Optimized Object Packings Using Quasi-Phi-Functions," has been authored by Yuriy Stoyan, Tatiana Romanova, Alexander Pankratov, and Andrey Chugay. The authors here further develop the main conceptual tool—called phi-functions—of their previous related studies. New quasi-phi-functions are defined and used for the analytical description of relations of geometric objects placed in a container taking into account their continuous rotations, translations, and distance constraints. These new functions are substantially simpler to use than phi-functions for certain types of objects. In particular, quasi-phi-functions are derived for certain two- and three-dimensional (2D and 3D) objects. The authors formulate a generic optimal packing problem and introduce its exact mathematical model as a continuous nonlinear programming problem, using quasi-phi-functions. Next, they propose a general solution strategy that includes the construction of feasible starting points; the generation of nonlinear sub-problems of a smaller dimension and smaller number of constraints; and the search for local extrema of the problem using sub-problems. To show the advantages of quasi-phi-functions, two packing problems are considered which have a broad spectrum of industrial applications. The first of these is the packing of a given collection of ellipses into a rectangular container of minimal area taking into account distance constraints. The second problem is the packing of a given collection of 3D objects—including cuboids, spheres, spherical cylinders, and spherical cones—into a cuboid container of minimal height. The authors developed efficient optimization algorithms to obtain locally optimal object packings. The algorithms are applied to solve several hard model instances, including both known and new test cases.

Chapter 14, titled "Graph Coloring Models and Metaheuristics for Packing Applications," has been written by Nicolas Zufferey. He considers and discusses the link between graph coloring and packings. In the classical graph coloring problem, a color has to be assigned to each vertex of a given graph. If two vertices are connected with an edge, then their colors have to be different. The goal is to color the graph with the smallest number of colors. Next, he considers the packing problem of loading items into containers: for each item, one has to decide the container assigned. Since by assumption certain pairs of items are incompatible, they cannot be loaded in the same container. The goal is then to load all the items in a minimum number of containers. Although the correspondence between these two problems is obvious (a vertex corresponds to an item, a color corresponds to a container, and a connecting edge represents an incompatibility), there is no apparent bridge between the packing and the graph coloring literatures. Several packing problems are formulated and solved applying graph coloring models and methods, and metaheuristics.

The broad range of OOP models, solution strategies, and applications discussed and presented by the contributing authors to this volume clearly illustrate the relevance of the subject. This book will be useful for researchers and practitioners in the field of OOP and numerous related fields. It will be useful also for graduate and post-graduate students to broaden their horizon, by studying real-world applications and challenging problems that they will meet in their professional work. Researchers and practitioners working in mathematical modeling, engineering design, operations research, mathematical programming, and optimization will benefit from the case studies presented. This book also offers extensive literature links for further studies: hence, it can be used as a reference source to assist researchers and practitioners in developing new OOP and related applications.

Turin, Italy Giorgio Fasano
Halifax, NS, Canada János D. Pintér
March 2015

http://www.springer.com/978-3-319-18898-0

Optimized Packings with Applications
fasano, g.; Pintér, J.D. (Eds.)
2015, XVII, 326 p. 95 illus., 67 illus. in color., Hardcover
ISBN: 978-3-319-18898-0