SIMULTANEOUS COLUMN-AND-ROW GENERATION FOR SOLVING LARGE-SCALE LINEAR PROGRAMS WITH COLUMN-DEPENDENT-ROWS

by

İBRAHİM MUTER

Submitted to the Graduate School of Engineering
and Natural Sciences in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Sabancı University

Spring 2011

SIMULTANEOUS COLUMN-AND-ROW GENERATION FOR SOLVING LARGE-SCALE LINEAR PROGRAMS WITH COLUMN-DEPENDENT-ROWS

APPROVED BY:

Ş. İLKER BİRBİL ....
(Thesis Supervisor)

KEREM BÜLBÜL ....

ANDREA LODI ....

SİBEL SALMAN ....

GÜVENÇ ŞAHİN ....

DATE OF APPROVAL: 28/07/2011

to my family

and

my deceased uncle Dr. Necmettin Erkan

SIMULTANEOUS COLUMN-AND-ROW GENERATION FOR SOLVING LARGE-SCALE LINEAR PROGRAMS WITH COLUMN-DEPENDENT-ROWS

İbrahim Muter

PhD Thesis, 2011

Thesis Advisor: Assoc. Prof. Dr. Ş. İlker Birbil

In this thesis, we handle a general class of large-scale linear programming problems. These problems typically arise in the context of linear programming formulations with exponentially many variables. The defining property for these formulations is a set of linking constraints, which are either too many to be included in the formulation directly, or the full set of linking constraints can only be identified, if all variables are generated explicitly. Due to this dependence between columns and rows, we refer to this class of linear programs as problems with *column-dependent-rows*. To solve these problems, we need to be able to generate both columns and rows on-the-fly within a new solution method. The proposed approach in this thesis is called *simultaneous column-and-row generation*. We first characterize the underlying assumptions for the proposed column-and-row generation algorithm. These assumptions are general enough and cover all problems with column-dependent-rows studied in the literature up until now. We then introduce, in detail, a set of pricing subproblems, which are used within the proposed column-and-row generation algorithm. This is followed by a formal discussion on the optimality of the algorithm. Additionally, this generic algorithm is combined with

Lagrangian relaxation approach, which provides a different angle to deal with simultaneous column-and-row generation. This observation then leads to another method to solve problems with column-dependent-rows. Throughout the thesis, the proposed solution methods are applied to solve different problems, namely, the multi-stage cutting stock problem, the time-constrained routing problem and the quadratic set covering problem. We also conduct computational experiments to evaluate the performance of the proposed approaches.

KOLON-BAĞLI-SATIR PROBLEMLERİNİN ÇÖZÜMÜ İÇİN EŞZAMANLI KOLON-VE-SATIR TÜRETME

İbrahim Muter

Doktora Tezi, 2011

Tez Danışmanı: Doç. Dr. Ş. İlker Birbil

Bu tezde genel bir problem sınıfa ait büyük-ölçekli doğrusal programlama problemleri ele alınmıştır. Bu problemler genellikle çok sayıda kolon içeren doğrusal programlarda ortaya çıkmaktadır. Bu formülasyonların ayırıcı özelliği bağlayıcı kısıtlardır. Bu kısıtlar ya formülasyona direk eklenemeyecek kadar çoktur ya da tüm kısıt seti ancak tüm kolonlar yaratıldığında tanımlanabilir. Kolon ve satırlar arasındaki bu bağımlılık nedeniyle bu doğrusal programlama sınıfına kolon-bağlı-satırlar problemleri denilmiştir. Bu problemleri çözebilmek için yeni bir çözüm yöntemi içinde hem kolon hem de satır türetilebilmelidir. Bu tezde önerilen çözüm yaklaşımına eşzamanlı kolon-ve-satır türetme adı verilmektedir. Öncelikle kolon-bağlı-satırlar problemleri için varsayımlar tanımlanmıştır. Bu varsayımlar yeterince geneldir ve literatürde bilinen tüm kolon-bağlı-satırlar problemlerini kapsamaktadır. Ardından önerilen kolon-ve-satır türetme algoritmasında kullanılan ücretlendirme altproblemleri detaylı olarak tanımlanmıştır. Bunu algoritmanın optimalliği üzerine formal bir tartışma izlemektedir. Ayrıca bu genel algoritma Lagrange gevşetmesi yaklaşımı ile birleştirilmiştir. Bu birleştirme kolon-ve-satır türetme için farklı bir bakış açısı sağladığı gibi kolon-bağlı-satırlar problemlerini çözmek için yeni bir yöntem ortaya koymaktadır. Önerilen çözüm yöntemleri çok-aşamalı stok kesme problemi, zaman-kısıtlı rotalama problemi ve karesel küme

kaplama problemi gibi değişik problemlere uygulanmıştır. Önerilen yaklaşımların performanslarını değerlendirmek için bilgisayısal deneyler yapılmıştır.

Acknowledgments

I thank my advisor Ş. İlker Birbil for his invaluable support during my Ph.D. study. It was and will be a pleasure to work with him. The most valuable lesson I learned from him is how to cultivate an idea for a scientific research. I am very thankful to Kerem Bülbül for his significant contribution to my research. I am very glad to work with Güvenç Şahin who gave me constant support and encouragement. I also thank the members of the research group AlgOpt.

I thank Andrea Lodi who introduced me to a new research area and made possible my research visit to Bologna. I also thank Sibel Salman for her helpful comments on my thesis.

I thank my officemates. It has always been fun to be in the same office with them. I am grateful to Mahir Yıldırım and Çetin Suyabatmaz for offering a space in their dorm room for me whenever necessary and for being great dudes. We shared the same anxiety and excitement with Figen Öztoprak and Taner Tunç as Ph.D. candidates. I thank both of them for their invaluable friendship. In numerous occasions, Taner Tunç cheered me up with his joy. I will remember my colleague Ezgi Yıldız with her laughters that made any boring working session enjoyable. Lastly, I thank Nurşen Aydın who took the courses I assisted at Marmara University and then became a colleague and a friend at Sabancı University.

Lastly, I owe everything I have achieved to my parents. Without their support, it would be very difficult to complete this work. I thank and apologize all those who bore with me during this long and painful period.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Linear programming (LP) deals with problems of maximizing or minimizing a linear function subject to a set of linear constraints. LP has been one of the most prominent tools used in the operations research field. One of the reasons is that LP problems have *nice* structures compared to the other optimization problems. Hence, these problems can be solved very efficiently. The major work on LP dates back to 1940s, when George Dantzig developed the simplex algorithm to solve LP problems [21]. The LP problem was first shown to be solvable in polynomial time by Khachiyan in 1979 [51], but a larger theoretical and practical breakthrough in the field came in 1984 when Karmarkar introduced a new interior point method for solving LP problems [49].

LP problems arise in diverse application areas. In many complex problems, such as; stochastic programming, nonlinear programming, combinatorial optimization, mixed integer programming problems and so on, LP is used as a modeling and solution tool. In the algorithms to solve these problems, LP problems are generally solved repeatedly and hence, the speed of the algorithms to solve LP problems becomes a major concern. The solution of an LP problem provides important information on the optimal solution of the original problem. To illustrate the use of LP, consider the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & c^\intercal x \\ \text{subject to} \quad & Ax \geq b, \\ & Bx \geq d, \\ & x \geq 0, \\ & x \text{ integer}, \end{aligned} \tag{1.1}$$

where A is an $m \times n$ matrix, B is a $k \times n$ matrix, and b, c, and d are $m \times 1$, $n \times 1$, and $k \times 1$ vectors, respectively. This problem is called the (linear) integer-programming problem. It is said to be a mixed integer program when some, but not all, variables are restricted to be integer, and is called a pure integer program when all decision variables must be integers. Since the objective function and the constraints are linear, the problem turns into an LP problem, if the integrality restrictions on the variables are ignored. It is well-known that the objective function value of the resulting LP problem provides a lower-bound on that of the original integer programming problem. In general, the optimal decision variables will be fractional in the linear-programming solution, and hence, further measures must be taken to determine an integer solution.

In this thesis and in many applications, we deal with LP problems with a large number of variables. Instead of solving these LP problems directly by an LP solver, various algorithms have been developed to find the optimal solution in a shorter computation time. Column generation is a prominent algorithm to solve large-scale LP problems.

When the number of variables is very large, it would even be impossible to enumerate all the variables in the problem. In such large-scale linear programs, the vast majority of the variables are zero at optimality. This is the fundamental concept underlying the column generation method, which is pioneered by Dantzig and Wolfe [23] as well as Gilmore and Gomory [39]. In this approach, the linear program is initialized with a small set of columns, referred to as the restricted master problem (RMP), and then new columns are added as required. This is accomplished iteratively by solving a pricing subproblem (PSP) following each optimization of the RMP. In the PSP, the reduced cost of a column is minimized over the set of all columns, and upon solving the PSP, we either add a new column to the RMP with a negative reduced cost (for minimization) or prove the optimality of the overall problem.

1.1 Motivations of This Research

One of the pillars of the classical column generation framework is that the constraints in the master problem are all known explicitly. In this case, the number of rows in the restricted master problem is fixed, and complete dual information is supplied to the PSP from the restricted master problem, which allows us to compute the reduced cost of a column in the subproblem accurately. While this framework has been used successfully for solving a large number of problems over the years, it does not fit applications in which missing columns induce new linking constraints to be added to the restricted master problem. To motivate the discussion, consider a quadratic set covering (QSC) model, where the binary variable y_k is set to 1, if column k is selected (see for example [66, 9]). We compute the total contribution from columns k and l as $c_k y_k + c_l y_l + c_{kl} y_k y_l$, where c_k and c_l are the individual contributions from columns k and l, respectively, and c_{kl} captures the cross-effect of having columns k and l simultaneously in the solution. A common linearization followed by relaxing the integrality constraints would lead to the large-scale LP below:

$$
\begin{aligned}
\text{minimize} \quad & \ldots + c_k y_k + c_l y_l + c_{kl} x_{kl} + \ldots \\
\text{subject to} \quad & \ldots \\
& y_k + y_l - x_{kl} \leq 1, \quad y_k - x_{kl} \geq 0, \quad y_l - x_{kl} \geq 0, \qquad (1.2) \\
& 0 \leq y_k, y_l, x_{kl} \leq 1, \\
& \ldots
\end{aligned}
$$

Note that this model contains three linking constraints for each pair of $y-$variables, and a large number of $y-$variables in an instance would prevent us from including all rows in the RMP a priori. Thus, in this case both rows and columns need to be generated on-the-fly as required. Constraints of type (1.2) not present in the current RMP may lead to two issues. First, primal feasibility may be violated with respect to the missing constraints. In order to address this issue, we should presumably add variable x_{kl} to the RMP along with one of the variables y_k or y_l. Second, the reduced costs of the variables may be computed incorrectly in the PSP because no dual information associated with the missing constraints is passed from the RMP to the PSP. For instance, assume that y_k is already a part of the RMP, while y_l, x_{kl}, and the linking constraints (1.2) are absent from it. In this case, the PSP for y_l must anticipate the values of the dual variables associated with the missing constraints (1.2); otherwise, the reduced cost of y_l is calculated incorrectly. Thus, we conclude that in order to design a column generation algorithm for this particular linearization of the quadratic set covering problem, we need a subproblem definition that allows us to generate several variables and their associated linking constraints simultaneously by correctly estimating the dual values of the missing linking constraints. Note that this type of dependence between columns can be generalized if several columns interact simultaneously and would lead to a similar problem that grows both column- and row-wise.

The discussion in the preceding paragraph points to a major difficulty in column generation, if the number of rows in the RMP depends on the number of columns. We refer to such formulations as problems with *column-dependent-rows*, or briefly as CDR-problems. We emphasize that the solution of a CDR-problem is based on *simultaneous column-and-row generation*. The cornerstone of this approach is a subproblem definition that can simultaneously generate new columns as well as new structural constraints that are no longer redundant in the presence of these new columns. This is in marked contrast to traditional column generation where all structural constraints are added to the RMP at the outset.

1.2 Contributions of The Thesis

In this dissertation, we introduce a column-and-row generation method that is able to overcome the difficulties resulting from the simultaneous addition of rows along with columns in a column generation method. We also study the combination of this method with Lagrangian relaxation.

To be more specific, first the problems that we refer to as CDR-problems are formulated. Then, a set of assumptions underlying these problems are defined, and the literature that deals with CDR-problems is discussed.

A generic column-and-row generation algorithm for CDR-problems is presented and the optimality of this algorithm is proved. The proposed approach is applied to the QSC, the multi-stage cutting stock (MSCS), and the time-constrained routing (TCR) problems. For the latter two problems, the existing methods are improved or in certain cases, corrected.

We also apply Lagrangian relaxation to CDR-problems by dualizing the set of linking constraints in the objective function. The resulting combination of column generation and Lagrangian relaxation is analyzed and then applied to the TCR problem.

This thesis is the first work in the literature, which addresses CDR-problems in a unified framework and gives a complete treatment of the optimality conditions along with associated optimal solution methods.

1.3 Outline of The Thesis

The current chapter is followed by Chapter 2, which includes a literature survey on the algorithms to solve large-scale LP problems. The main focus will be on column generation. Since the problems that we discuss are integer programming problems, we also give a survey on the lower-bounding methods and the ways to find the optimal integral solutions for general classes of integer programming problems. Finally, the literature related specifically to the CDR-problems are presented.

In Chapter 3, the generic mathematical model and the underlying assumptions for the CDR-problems are given. We demonstrate that two of the CDR-problems that we use as illustrative examples, namely the MSCS and QSC problems, conform to our generic mathematical model and the underlying assumptions. We also emphasize that all the CDR-problems in the literature can be cast into our generic model, and they all satisfy our assumptions.

In Chapter 4, the proposed simultaneous column-and-row generation method is presented. Then, the optimality proof of this method is given. After this presentation, the proposed method is applied to the MSCS, QSC, and TCR problems. Computational experiments are then conducted on the MSCS and QSC problems to evaluate the performance of the algorithm.

In Chapter 5, the combination of the column-and-row generation algorithm with Lagrangian relaxation is investigated, and its differences from the simultaneous column-and-row generation method given in Chapter 4 are pointed out. The resulting hybrid method is applied to the TCR problem, and it is compared against the column-and-row generation algorithm of Chapter 4 by conducting additional computational experiments.

In the last chapter, we summarize the conclusions of this dissertation and discuss further research directions.

Chapter 2

LITERATURE REVIEW

In this chapter, first we present methods which are used to solve large-scale LP problems. Since the problems we deal with in this thesis are integer programming and combinatorial optimization problems whose particular relaxation is an LP problem, we explain algorithms to find an integral optimal solution. Moreover, the literature related specifically to the CDR-problems is presented.

2.1 Large-Scale Linear Programming Problems

The advances in the solution algorithms for large-scale LP problems have made these problems a viable relaxation to many difficult problems. In particular, large-scale integer programming and combinatorial optimization problems depend on the solution of large-scale LP problems. In this section, we give a survey on the algorithms, particularly the column generation algorithm and the Benders decomposition algorithm, that are used to solve large-scale LP problems. We explain the column generation algorithm as a general idea to solve the Dantzig-Wolfe decomposition problem (see [17] for the details). Benders decomposition takes place in this review since it can also be applied to the CDR-problems.

Column Generation. Consider the following LP problem which is the relaxation of (1.1) given in Chapter 1:

$$\begin{aligned} \text{minimize} \quad & c^\intercal x, \\ \text{subject to} \quad & Ax \geq b, \\ & Bx \geq d, \\ & x \geq 0. \end{aligned} \tag{2.1}$$

Suppose that (2.1) is significantly easier to solve when the set of constraints $Ax \geq b$ is removed. This could be, for instance, because the resulting problem after removal is easy to decompose into smaller independent problems. In fact, such problems are said to have block diagonal structure. In this setting, the set of constraints $Ax \geq b$ is often called the *complicating constraint* set.

Let us denote the polyhedron induced by the second set of constraints and the nonnegativity constraints as $\mathcal{P} = \{x \in \mathbb{R}^n | Bx \geq d, x \geq 0\} \neq \emptyset$. Using the representation theorem of Minkowski [62], any point in $\mathcal{P}$ can be represented by the convex combination of its extreme points $\{p_q\}_{q \in Q}$ plus a nonnegative combination of its extreme rays $\{p_r\}_{r \in R}$ of $\mathcal{P}$. Hence, the representation of any point $x \in \mathcal{P}$ in terms of the extreme points and the rays is given by

$$x = \sum_{q \in Q} p_q \lambda_q + \sum_{r \in R} p_r \lambda_r, \ \sum_{q \in Q} \lambda_q = 1, \ \lambda \in \mathbb{R}_+^{|Q|+|R|}. \tag{2.2}$$

Substituting for x in (2.1) and applying the linear transformations $c_q = c^\intercal p_q, \ q \in Q$ and $a_r = Ap_r, \ r \in R$, we obtain an equivalent extensive formulation of (2.1):

$$\begin{aligned} \text{minimize} \quad & \sum_{q \in Q} c_q \lambda_q + \sum_{r \in R} c_r \lambda_r, \\ \text{subject to} \quad & \sum_{q \in Q} a_q \lambda_q + \sum_{r \in R} a_r \lambda_r \geq b, \\ & \sum_{q \in Q} \lambda_q = 1, \\ & \lambda \geq 0, \lambda \in \mathbb{R}_+^{|Q|+|R|}. \end{aligned} \tag{2.3}$$

Typically, problem (2.3) has a large number of variables ($|Q| + |R|$), but possibly substantially fewer rows than problem (2.1). The second constraint is referred to as the convexity constraint over the extreme points of $\mathcal{P}$. This substitution is known as Dantzig-Wolfe decomposition, developed in [23], and it easily generalizes to the case when matrix B is block diagonal.

As we pointed out previously, a column generation algorithm is generally used when the number of columns in the problem is very large. It is either applied to the extensive formulations (Dantzig-Wolfe decomposition) of problems with a set of complicating constraints or to compact (original) formulations with (exponentially) many columns. The compact formulation and the extensive formulation are structurally similar except for the convexity constraints.

The RMP, which is formed by a subset of the columns of (2.3) indexed by $\bar{Q}$ and $\bar{R}$, is given by

$$\begin{aligned} \text{minimize} \quad & \sum_{q\in\bar{Q}} c_q\lambda_q + \sum_{r\in\bar{R}} c_r\lambda_r, \\ \text{subject to} \quad & \sum_{q\in\bar{Q}} a_q\lambda_q + \sum_{r\in\bar{R}} a_r\lambda_r = b, \\ & \sum_{q\in\bar{Q}} \lambda_q = 1, \\ & \lambda \geq 0, \lambda \in \mathbb{R}_+^{|\bar{Q}|+|\bar{R}|}. \end{aligned} \tag{2.4}$$

Let α and β be the optimal dual variables corresponding to the first and the second constraints of the RMP given in (2.3), respectively. The corresponding PSP is then equivalent to

$$\begin{aligned} \text{minimize} \quad & (c^\intercal - \alpha^\intercal A)x - \beta, \\ \text{subject to} \quad & Bx \geq d, \\ & x \geq 0. \end{aligned} \tag{2.5}$$

The main objective of this model is to find the columns with the minimum reduced costs. If the minimum reduced cost is negative and finite, a column corresponding to an extreme point is added to the model. If the minimum reduced cost is negative and infinite, a column corresponding to an extreme ray is added to the model. Otherwise, the algorithm terminates. The typical flow of a column generation algorithm is outlined in Figure 2.1. The novel applications of column generation to integer programming problems include [27, 24, 64, 6, 36, 69, 1] (see also [25, 55] for comprehensive surveys on column generation).

Figure 2.1: The flow of a typical column generation algorithm.

The RMP is generally solved by the simplex algorithm from which we obtain the optimal dual variables to be used in the PSP. Unfortunately, the convergence of the simplex algorithm may be poor. One of the reasons is degeneracy which results in many iterations without improvement. Additionally, the dual solution oscillates dramatically during the early phases of the algorithm and this may add many useless columns (see [68] for the related issues). As a remedy, stabilized column generation algorithms have been proposed. One approach studied in [29] perturbs the right hand side to reduce degeneracy and uses a box concept to limit the variation in the dual variables. Interior point methods, such as the analytic center method [30], have also been used to solve the RMP.

Benders Decomposition. Benders decomposition developed in [12] is useful for solving problems that contain groups of variables of different natures. While Dantzig-Wolfe decomposition deals with the complicating constraints, Benders decomposition handles the problems with complicating variables whose removal results in a significantly easier problem. Hence, it is a dual idea with respect to Dantzig-Wolfe decomposition. There are many applications of this methodology to mixed-integer programming problems. Some examples are the multi-commodity distribution network design, the locomotive and car assignment, the simultaneous aircraft routing and crew scheduling, and the large scale water resource management problems [37, 19, 18, 13].

The basic model, where x is taken as the complicating variable set, is given by

$$\begin{aligned} \text{minimize} \quad & c^\intercal x + f^\intercal y \\ \text{subject to} \quad & Ax \geq b, \\ & Bx + Dy \geq d, \\ & x \geq 0, y \geq 0, \end{aligned} \tag{2.6}$$

where c, A, B, b and d are defined as in (1.1), f is a $l \times 1$ vector, and D is a $k \times l$ matrix. Reformulating this model leads to the following two level structure

$$\begin{aligned} \text{minimize} \quad & c^\intercal x + z(x), \\ \text{subject to} \quad & Ax \geq b, \\ & x \geq 0, \end{aligned} \tag{2.7}$$

where

$$\begin{aligned} z(x) = \text{minimize} \quad & f^\intercal y, \\ \text{subject to} \quad & Dy \geq d - Bx, \\ & y \geq 0. \end{aligned} \tag{2.8}$$

Given the value of x and applying duality, the optimal solution of problem (2.8) can be obtained by solving

$$\begin{aligned} z(x) = \text{maximize} \quad & (d - Bx)^\intercal v, \\ \text{subject to} \quad & Dv \leq f, \\ & v \geq 0, \end{aligned} \tag{2.9}$$

where v is the set of dual variables corresponding to the first set of constraints in (2.8). Note that the dual polyhedron, which we will denote by Φ, is independent of x. Therefore, using the representation theorem, we can enumerate the set of extreme points and extreme rays of Φ as $P_\Phi = \{p_1, p_2, ..., p_P\}$ and $Q_\Phi = \{q_1, q_2, ..., q_Q\}$. Using these extreme points and extreme rays, problem (2.8) can be written as

$$\begin{aligned} z(x) = \text{minimize} \quad & z, \\ \text{subject to} \quad & (d - Bx)^\intercal v \leq z, \quad v \in P_\Phi, \\ & (d - Bx)^\intercal v \leq 0, \quad v \in Q_\Phi, \end{aligned} \tag{2.10}$$

which finds the dual extreme point resulting in the maximum objective function value.

If we plug in $z(x)$ in (2.7), the resulting problem then becomes

$$\begin{aligned} \text{minimize} \quad & c^{\intercal}x + z, \\ \text{subject to} \quad & Ax \geq b, \\ & (d - Bx)^{\intercal}v \leq z, \quad v \in P_{\Phi}, \\ & (d - Bx)^{\intercal}v \leq 0, \quad v \in Q_{\Phi}, \\ & x \geq 0. \end{aligned} \tag{2.11}$$

When compared to (2.6), problem (2.11) has fewer variables, since variables y do not exist in this problem. However, the number of constraints in (2.11) is considerably larger, since there is one constraint for each extreme point and extreme ray.

Enumerating all extreme points and extreme rays may be very time-consuming. Therefore, we may instead include only a subset of the constraints corresponding to the sets of extreme points and extreme rays and add the violated constraints on the fly. This approach is known as *delayed constraint generation*. The restricted Benders master problem (BMP) at any iteration t is given by

$$\begin{aligned} Z^t = \text{minimize} \quad & c^{\intercal}x + z, \\ \text{subject to} \quad & Ax \geq b, \\ & (d - Bx)^{\intercal}v \leq z, \quad v \in P_{\Phi}^t, \\ & (d - Bx)^{\intercal}v \leq 0, \quad v \in Q_{\Phi}^t, \\ & x \geq 0, \end{aligned} \tag{2.12}$$

where P_{Φ}^t and Q_{Φ}^t are subsets of P_{Φ} and Q_{Φ}, respectively. The optimal objective function value and the optimal solution are denoted by Z^t and $(\overline{x}, \overline{z})$, respectively. Using the $\overline{x}$ values, the dual subproblem (SP) in (2.9) is solved. If the solution of SP is unbounded, letting $\overline{v}$ be the corresponding extreme ray, we add the feasibility cut to the BMP as a constraint given by

$$(d - Bx)^{\intercal}\overline{v} \leq 0. \tag{2.13}$$

If there is an optimal solution to SP, an extreme point is obtained. If we denote this extreme point by $\overline{v}$, then we check whether

$$(d - B\overline{x})^{\intercal}\overline{v} > \overline{z} \tag{2.14}$$

holds. If (2.14) holds, then the optimality cut corresponding to the extreme point $\overline{v}$ is added to the BMP as a constraint given by

$$(d - Bx)^{\intercal}\overline{v} \leq z. \tag{2.15}$$

Otherwise; i.e., if no constraint is violated, the algorithm terminates. An important extension to this algorithm is presented in [38] which suggests a generalized Benders decomposition approach. In this study, the Benders method is extended to the case where the subproblem is a convex optimization problem. In [58], the influence of cuts in a Benders Decomposition algorithm applied to mixed integer programs is studied and, a new technique for accelerating the convergence of the algorithm through model formulations and selection of Pareto-optimal cuts are introduced.

2.2 Integer Programming Problems

Suppose that the polyhedron $\mathcal{P} = \{x \in \mathbb{R}^n | Bx \geq d, x \geq 0\} \neq \emptyset$ given earlier is replaced by a finite set $X = \mathcal{P} \cap Z_+$. Then, problem (2.1) becomes

$$\begin{aligned} \text{minimize} \quad & c^{\intercal}x \\ \text{subject to} \quad & Ax \geq b, \\ & x \in X, \end{aligned} \tag{2.16}$$

which is known to be an NP-Complete optimization problem [62]. In this section, we discuss the branch-and-bound algorithm, which is by far the most widely used tool for solving large scale NP-hard combinatorial optimization problems. The bounding operation can be performed by several algorithms such as: LP relaxation (with column generation), Lagrangian relaxation and cutting plane algorithms.

Branch-and-Bound Algorithm. The branch-and-bound method is based on the idea of iteratively partitioning the set of feasible solutions to form subproblems of the original integer program that are easier to handle. The same process is applied to the subproblems, and this process goes on until the optimal solution of any subproblem provides an optimal solution to the original problem. This is called branching and as the number of branches increases, the number of subproblems grows exponentially. Hence, it becomes crucial to eliminate some of the subproblems. This requires a bounding scheme, which is based on setting lower and upper bounds on the optimal objective function values of the subproblems. The relaxations of the subproblems are generally solved to speed-up the algorithm and any feasible solution to the original problem obtained by solving a subproblem gives an upper bound. The best upper bound is recorded during the search. The driving force behind the branch-and-bound approach lies in the fact that if a lower bound for the objective value of a given subproblem is larger than the best upper bound, then the optimal solution of the original integer program cannot lie in the subset of solutions associated with the given subproblem. Hence, the corresponding subset is pruned. Hence, the lower bounds on the objective function values of the subproblems are, in essence, used to construct a proof of optimality without doing an exhaustive search of the branches.

To implement the branch-and-bound algorithm, several decisions must be made. Among these decisions we can give the two most prominent ones as examples; the branching and the subproblem selection strategies (see [62] for the details). To approximate the optimal solution of the integer program defined on a node of the branch-and-bound tree, different bounding procedures are used. Most bounding procedures are based on the generation of a polyhedron that approximates the convex hull of feasible solutions. Solving an optimization problem over such a polyhedral approximation produces a bound that can be used in a branch and bound algorithm. The effectiveness of the bounding procedure depends largely on how well X in (2.16) can be approximated. The most straightforward approximation is the continuous approximation, which boils down to the LP relaxation. The bound resulting from this approximation is frequently weak. The success of the other bounding algorithms given below relies heavily on the effectiveness of the solution methodology to solve the PSP, Lagrangian subproblem or the separation subproblem. Heuristic algorithms are frequently employed to solve these subproblems as long as a column or cut that improves the bound is detected. Otherwise, they must be solved exactly to prove optimality. Two examples, where such a strategy is used, are the pick-up and delivery problem [65] and the capacitated vehicle routing problem [57]. Next, we discuss the Lagrangian relaxation and the cutting plane methods that can be used to find a lower-bound at a node of the branch-and-bound tree.

Lagrangian Relaxation. Lagrangian relaxation is widely employed to obtain an improved bound (see [34] for the details of the use of Lagrangian relaxation in IP). Lagrangian relaxation algorithm moves the complicating constraint set in (2.16) to the objective function by multiplying it with the Lagrangian multiplier vector u. The resulting problem becomes

$$L(u) := \min_{x \in X} c^{\intercal} x - u^{\intercal}(Ax - b). \tag{2.17}$$

Given vector u, the optimal objective function value, $L(u)$, provides a lower-bound on the optimal solution of the integer program. Clearly, the best bound then can be obtained by solving the Lagrangian dual problem given by

$$\max_{u \geq 0} L(u). \tag{2.18}$$

In [62], the dual of the Lagrangian dual model is shown to be equivalent to the Dantzig-Wolfe decomposition for a given X. It is well-known that Lagrangian relaxation is obtained by dualizing exactly those constraints that are the linking constraints in the Dantzig-Wolfe decomposition. Moreover, the subproblem that we need to solve in the column generation procedure is the same as the one we have to solve for the Lagrangian relaxation except for a constant term in the objective function. Column generation and Lagrangian relaxation provide the same bounds which is better than the LP relaxation, if the convex hull of X does not have integrality property. Hence, the constraints to be relaxed are generally selected to violate the integrality property. To find an upper bound, either Lagrangian relaxation is embedded in a branch-and-bound algorithm as in [4, 53] or a Lagrangian heuristic approach is employed as in [14, 10].

There are several methods to solve the Lagrangian dual problem given in (2.18). Subgradient algorithm is widely used to find the optimal Lagrangian multipliers in (2.18). This algorithm is first applied in the Lagrangian relaxation context to the traveling salesman problem by Held and Karp [43, 44]. The performance and the theoretical convergence properties of subgradient optimization are given in [45]. Other successful alternatives to the subgradient optimization are the volume algorithm and the bundle algorithm (see [5, 54], respectively, for the details of these algorithms).

Cutting Plane Algorithms. Cutting plane methods improve the continuous approximation by dynamically generating valid inequalities to form a better approximation of the convex hull of the feasible region. An inequality $\pi x \leq \pi_0$ is a valid inequality for X if $\pi x \leq \pi_0$ for all $x \in X$. The valid inequalities are generated by solving a separation problem. The addition of each valid inequality cuts the approximating polyhedron, resulting in a potentially improved bound.

The general cutting plane algorithm solves the continuous relaxation of the problem and checks if the optimal solution violates any of the valid inequalities by solving a separation subproblem. If this is the case, the most violated valid inequality is added; otherwise, the algorithm terminates. If the solution is not integral, then branching takes place.

The cuts can be classified as general cutting planes and cuts for special structures. Gomory cuts are one of the most prominent general classes of cutting planes [40]. These cuts can be applied to any integer linear program. However, exploiting the special structure of the given problem may result in more effective cuts. Such cuts exploiting the special structure of the problem have been successfully used in the traveling salesman problem (see [22, 20, 63] for different cuts).

Depending on the selected bounding algorithm, the branch-and-bound algorithm takes different names. To obtain an integral optimal solution using column generation, it must be embedded in a branch-and-bound framework. This procedure is called branch-and-price [27, 8]. In this case, new columns are generated at each node of the branch-and-bound tree and branching is implemented when no columns enter the basis and the LP relaxation is fractional. When the cutting plane procedure is applied at each node of the branch-and-bound tree, the resulting procedure is called branch-and-cut [46, 62]. When the cuts do not result in an integral solution, branching occurs. Branch-and-cut-and-price, on the other hand, employs column generation and cut generation at each node of the branch-and-bound tree together [7, 11].

2.3 Existing Work on Problems with Column-Dependent-Rows

The literature on the CDR-problems is somewhat limited. In this section, we discuss the existing work in the literature and position our contributions. When it comes to the CDR-problems mentioned in this section, it is relatively easy to check that these problems satisfy our assumptions that will be defined in the next chapter. Therefore, the proposed column-and-row generation algorithm indeed provides a generic approach to solve these problems.

To the best of our knowledge, the first column-and-row generation algorithm as we consider here was devised in [73], who tries to solve a one-dimensional MSCS problem. The algorithm developed in [73] is based on a restrictive assumption, which causes the algorithm to terminate at a suboptimal solution. A two-stage batch scheduling problem that is structurally similar to MSCS is formulated in [71] and the proposed algorithm suffers from an analogous restrictive assumption. MSCS will be introduced in Section 3.2 and our solution method will be applied to this problem in Section 4.2.1.

In [2], a time-constrained routing (TCR) problem motivated by an application that needs to schedule the visit of a tourist to a given geographical area as efficiently as possible in order to maximize her total satisfaction is studied. The goal is to send the tourist on one tour during each day in the vacation period while ensuring that each attraction site is visited no more than once. This problem is formulated as a set packing problem with side constraints and solved heuristically by a column-and-row generation approach due to a potentially huge number of tours. The authors enumerate and store a large number of tours before invoking their column generation algorithm. The SRMP for solving the LP relaxation of the proposed formulation is initialized with a subset of the enumerated tours. A selected tour must be assigned to one of the days in the vacation period. Each generated tour during the column generation procedure introduces a set of variables and leads to a new linking constraint in the SRMP. The authors define an optimality condition for terminating their column generation algorithm based on the dual variables of the constraints in the current SRMP. Following each optimization of the SRMP, this condition is verified for each tour currently absent from the SRMP; i.e., no PSP is required. In Section 4.2.3, we demonstrate that this stopping condition fails to account for the dual variables of the missing linking constraints properly and may lead to a suboptimal LP solution at termination.

A branch-and-cut-and-price algorithm for the well-known P-median problem is proposed in [3]. In their formulation, a set of binary variables indicate the set of selected median nodes, and binary assignment variables designate the median node assigned to each node in the network. These two types of binary variables are linked by variable upper bound constraints. One of the main contributions of the authors is a column-and-row generation method for solving the LP relaxation of this formulation. The algorithm is invoked with a subset of the assignment variables and additional ones are generated as necessary. The generation of each assignment variable leads to a single new linking constraint added to the SRMP for primal feasibility, and the dual variable associated with this linking constraint is calculated correctly a priori due to the special structure of the formulation and incorporated directly into the reduced cost calculations. No PSP is required because all potential assignment variables are known explicitly. Similar to the formulation in the previous work of these authors on the TCR problem, we note that the P-median formulation investigated in [3] is a special case of our generic formulation (MP) and can be handled by our proposed solution methodology.

In [61], a robust airline crew pairing problem for managing extra flights with the objective of hedging against a certain type of operational disruption by incorporating robustness into the pairings generated at the planning level is studied. In particular, they address how a set of extra flights may be added into the flight schedule at the time of operation by modifying the pairings at hand and without delaying or canceling the existing flights in the schedule. Essentially, this is accomplished in two different ways. An extra flight may either be inserted into an existing pairing with ample connection time (a type-B solution) or the schedules of a pair of pairings are partially swapped to cover an extra flight while ensuring the feasibility of these two pairings before and after the swap (a type-A solution). In the latter case, there is a benefit of having a pair of pairings in the solution simultaneously. However, an additional complicating factor is that the set of type-A solutions and the associated linking constraints are not known explicitly. This is akin to the MSCS problem, where the set of intermediate rolls is not available a priori. Ultimately, the mathematical model proposed in [61] boils down to a QSC problem with restricted pairs and side constraints. The model is linearized by the same approach as that in (3.6)-(3.12) for the QSC problem. A heuristic two-phase iterative column-and-row generation strategy is devised in [61] to solve the LP relaxation of their master problem. In the first phase, the number of constraints in the SRMP is fixed and column generation is applied in a classical manner. Then, in the second phase additional type-A solutions are identified based on the pairings generated during the last call to the column generation with a fixed number of constraints, and the associated constraints are added to the SRMP before the next iteration of the algorithm resumes. We note that the problem in [61] is a CDR-problem and can be handled by the proposed methodology in this thesis.

Finally, we refer to a recent work in [32]. In this work, the optimality conditions for column-and-row generation are analyzed for two sample problems; the split delivery vehicle routing problem and the service network design problem for an urban rapid transit system. The authors claim that there is no simple rule to construct an optimal solution and thus, one has to define specifically how to proceed for every application case. Our work, however, does state a generic model and characterizes the type of problems that can be solved by column-and-row generation including those discussed in [32]. Besides, we also propose an associated solution framework to design a column-and-row generation algorithm for CDR-problems.

Column-and-row generation (or row-and-column generation) is a term without a widely-agreed precise definition. Therefore, we conclude this section by distinguishing our work from others, who use the same term in a different context. For instance, in both [35] and [50], the multi-commodity capacitated network design problem is considered and column-and-row generation algorithms are employed. In both of these cases, the rows that are added to the formulation are valid inequalities that strengthen the LP relaxation in line with the general branch-and-cut-and-price paradigm (see [26, 28]). This is very different than our framework for CDR-problems, in which generated rows are structural constraints that are required for the validity of the formulation. Furthermore, as pointed out in [35] the column- and row generation subproblems in the branch-and-cut-and-price context are either independent from each other or generated columns introduce new cuts with trivial separation problems. For a CDR-problem, the situation is completely different as we study thoroughly in Chapter 4.

Chapter 3

PROBLEMS WITH COLUMN-DEPENDENT-ROWS

In this chapter, we first specify the canonical form of the generic mathematical model representing the class of CDR-problems that we consider. Then, we discuss the assumptions underlying our modeling and solution framework. To illustrate our construction, we briefly describe two example problems, the MSCS and QSC problems, and demonstrate that both of these problems satisfy our assumptions and they may conform to our generic model. These two problems are selected for their different characteristics that help us illustrate the different features and aspects of our proposed solution method.

3.1 Generic Mathematical Model

The generic mathematical formulation of CDR-problems appears below, and we refer to it as the *master problem*, following the common terminology in column generation:

$$
\begin{array}{llrlr}
(\mathbf{MP}) & \text{minimize} & \displaystyle\sum_{k \in K} c_k y_k + \sum_{n \in N} d_n x_n, & & \\
& \text{subject to} & \displaystyle\sum_{k \in K} A_{jk} y_k & \geq a_j, \quad j \in J, & \text{(MP-y)} \\
& & \displaystyle\sum_{n \in N} B_{mn} x_n & \geq b_m, \quad m \in M, & \text{(MP-x)} \\
& & \displaystyle\sum_{k \in K} C_{ik} y_k + \sum_{n \in N} D_{in} x_n & \geq r_i, \quad i \in I, & \text{(MP-yx)} \\
& & y_k \geq 0, k \in K, \quad x_n \geq 0, n \in N. & &
\end{array}
$$

There may be exponentially many $y-$ and $x-$ variables in this formulation, and we allow both types of variables to be generated in a column generation algorithm applied to solve the master problem. We assume that the set of constraints (MP-y) and (MP-x) are known explicitly and their cardinality is polynomially bounded in the size of the problem. On the other hand, a complete description of the set of linking constraints (MP-yx) may not be available. If this is the case, we may have to generate all $y-$ and $x-$ variables in the worst case to identify all linking constraints in a column generation algorithm. The discussion on a robust crew pairing problem studied in [61] in Section 2.3 provides an example for this case. Even if all linking constraints (MP-yx) are known explicitly a priori, there may be exponentially many of them. For instance, in the QSC example introduced in the previous section each pair of variables induces three linking constraints in the linearized formulation, and incorporating all $\mathcal{O}(\mid K \mid^2)$ linking constraints in the formulation directly is not a viable alternative for large $\mid K \mid$.

Based on the discussion in the preceding paragraph, the column-and-row generation algorithm for solving the master problem is initialized with subsets $\bar{K} \subset K$ and

$\bar{N} \subset N$. The resulting model then becomes

$$
\begin{array}{llllll}
(\textbf{SRMP}) & \text{minimize} & \sum_{k\in\bar{K}} c_k y_k + & \sum_{n\in\bar{N}} d_n x_n, & & \\
& \text{subject to} & \sum_{k\in\bar{K}} A_{jk} y_k & \geq a_j, & j \in J, & \text{(SRMP-y)} \\
& & & \sum_{n\in\bar{N}} B_{mn} x_n \geq b_m, & m \in M, & \text{(SRMP-x)} \\
& & \sum_{k\in\bar{K}} C_{ik} y_k + & \sum_{n\in\bar{N}} D_{in} x_n \geq r_i, & i \in I(\bar{K},\bar{N}), & \text{(SRMP-yx)} \\
& & y_k \geq 0, k \in \bar{K}, \quad x_n \geq 0, n \in \bar{N}, & & &
\end{array}
$$

where $I(\bar{K},\bar{N}) \subset I$ in (SRMP-yx) denotes the set of linking constraints formed by $\{y_k|k \in \bar{K}\}$, and $\{x_n|n \in \bar{N}\}$. During the column generation phase, new variables $\{y_k|k \in S_K\}$ and $\{x_n|n \in S_N\}$, where $S_K \subset (K \setminus \bar{K})$ and $S_N \subset (N \setminus \bar{N})$, are added to the RMP iteratively as required as a result of solving different types of PSPs which we discuss in depth in Section 4.1. Moreover, these new variables may appear in new linking constraints currently absent from the RMP, where the set of these new linking constraints is represented by $\Delta(S_K, S_N) = I(\bar{K} \cup S_K, \bar{N} \cup S_N) \setminus I(\bar{K},\bar{N})$. Thus, the RMP grows both vertically and horizontally during column generation, and due to this special structure we refer to the RMP in our column-and-row generation algorithm as the *short restricted master problem* (SRMP).

Three main assumptions characterize the type of problems that fit into our generic model and that we can tackle by our proposed solution methodology. In the next section, we argue that all of these assumptions hold for our two illustrative CDR-problems; QSC and MSCS. Moreover, in Section 2.3 we considered other problems from the literature, for which it is trivial to check that these assumptions also apply. The first assumption implies that the generation of the $x-$variables depends on the generation of the $y-$variables. Moreover, each $x-$variable is associated with only one set of linking constraints.

Assumption 3.1.1 *The generation of a new set of variables* $\{y_k|k \in S_K\}$ *prompts the generation of a new set of variables* $\{x_n|n \in S_N(S_K)\}$. *Furthermore, a variable*

$x_{n'}, n' \in S_N(S_K)$*, does not appear in any* linking constraints *other than those indexed by* $\Delta(S_K, S_N(S_K))$ *and introduced to the SRMP along with* $\{y_k | k \in S_K\}$ *and* $\{x_n | n \in S_N(S_K)\}$*.*

Note that the dependence of $\bar{N}$ on $\bar{K}$ is designated by the index set $S_N(S_K)$. In the remainder of the thesis, we will use the shorthand notation $\Delta(S_K)$ instead of $\Delta(S_K, S_N(S_K))$ whenever there is no ambiguity.

The next assumption requires the definition of a *minimal variable set*. A minimal variable set is a set of $y-$variables that triggers the generation of a set of $x-$variables and the associated linking constraints in the sense of Assumption 3.1.1. In the QSC formulation in Section 1.1, a minimal variable set given by $\{y_k, y_l\}$ consists of the variables y_k and y_l and generates a set of linking constraints of type (1.2) and the variable x_{kl}. We also note that in our subsequent discussion, we shall see that there may be several minimal variable sets associated with a set of linking constraints. Thus, we state the following assumption for the general case.

ASSUMPTION 3.1.2 *A linking constraint is redundant until* all *variables in at least one of the minimal variable sets associated with this linking constraint are added to the SRMP.*

This assumption implies that a feasible solution of SRMP does not violate any missing linking constraint before all variables in at least one of the associated minimal variable sets are added to the SRMP.

Assumptions 3.1.1 and 3.1.2 together define the goal of the fundamental subproblem in our proposed column-and-row generation approach. The objective of the *row-generating PSP* derived in Section 4.1 is to identify one or several minimal variable sets, where each minimal variable set $\{y_k | k \in S_K\}$ yields a set of variables $\{x_n | n \in S_N(S_K)\}$. These two sets of variables appear in a set of linking constraints indexed by $\Delta(S_K)$ currently not present in the SRMP, and we are also required to add these constraints to the SRMP to avoid violating the primal feasibility of the master problem (MP). Thus, for each new minimal variable set $\{y_k | k \in S_K\}$ to be introduced into the SRMP as an output of the row-generating PSP, the index sets defining SRMP are updated as

$\bar{K} \leftarrow \bar{K} \cup S_K$, $\bar{N} \leftarrow \bar{N} \cup S_N(S_K)$, and a new set of constraints $\Delta(S_K)$ appear in the SRMP. Clearly, at least one of the currently generated $y-$variables must have a negative reduced cost.

The next assumption characterizes the signs of the coefficients in the linking constraints.

ASSUMPTION 3.1.3 *Suppose that we are given a minimal variable set* $\{y_l | l \in S_K\}$ *that generates a set of linking constraints* $\Delta(S_K)$ *and a set of associated* $x-$*variables* $\{x_n | n \in S_N(S_K)\}$*. When the set of linking constraints* $\Delta(S_K)$ *is* first *introduced into the SRMP during the column-and-row generation, then for each* $k \in S_K$ *there exists a constraint* $i \in \Delta(S_K)$ *of the form*

$$C_{ik} y_k + \sum_{n \in S_N(S_K)} D_{in} x_n \geq 0, \tag{3.1}$$

where $C_{ik} > 0$ *and* $D_{in} < 0$ *for all* $n \in S_N(S_K)$*.*

Assumption 3.1.3 ensures that a variable $x_n, n \in S_N(S_K)$, cannot assume a positive value until *all* variables in at least one of the minimal variable sets that generate $\Delta(S_K)$ are positive in the SRMP. In addition, we emphasize that although we use (3.1) throughout this thesis, our analysis is also valid when a constraint of type (3.1) is given in a disaggregated form like

$$C_{ik} y_k + D_{in} x_n \geq 0, \quad n \in S_N(S_K).$$

Furthermore, linking constraints of type (3.1) may be specified as equalities in some CDR-problems. This case may also be handled with minor modifications to the analysis in Section 4.1. An example of the equality case can be found in Section 4.2.3 where our proposed approach is applied to the TCR problem.

We further classify CDR-problems as *CDR-problems with interaction* and *CDR-problems with no interaction*. This distinction between two problem types plays an important role in our analysis.

Definition 3.1.1 *In a CDR-problem* with interaction*, the cardinality of any minimal variable set is larger than one. On the other hand, if each minimal variable set is a singleton, then the corresponding problem belongs to the class of CDR-problems* with no interaction.

Differentiating between CDR-problems with and with no interaction allows us to focus on the unique properties of these two types that affect the analysis of the row-generating PSP in Section 4.1. However, it is possible to combine the tools developed in this thesis to tackle CDR-problems in which some minimal variable sets are singletons while others include more than one variable. This extension is discussed in Section 4.3.

3.2 Illustrative Examples

In the one-dimensional multi-stage cutting stock (MSCS) problem, operational restrictions impose that stock rolls are cut into finished rolls in more than one stage (see [42, 33, 72, 73]). The objective is to minimize the number of stock rolls used for satisfying the demand for finished rolls, and appropriate cutting patterns need to be identified for each stage in the cutting process. We restrict our attention to the two-stage cutting stock problem similar to the study by [73]. In the first stage, a stock roll is cut into intermediate rolls, while finished rolls are produced from these intermediate rolls in the second stage. If we ignore the integrality restrictions, then the LP model for the MSCS problem is given by

$$\text{minimize} \quad \sum_{k\in K} y_k, \tag{3.2}$$

$$\text{subject to} \quad \sum_{n\in N} B_{mn}x_n \geq b_m, \quad m\in M, \tag{3.3}$$

$$\sum_{k\in K} C_{ik}y_k + \sum_{n\in N} D_{in}x_n \geq 0, \quad i\in I, \tag{3.4}$$

$$y_k \geq 0, k\in K, \quad x_n \geq 0, n\in N, \tag{3.5}$$

where the set of intermediate and finished rolls are denoted by I and M, respectively. The set of cutting patterns K for the first stage constitute the columns of C. Similarly, the columns of B are formed by the set of cutting patterns N for the second stage. The matrix D establishes the relationship between the cutting patterns in the first and the second stages. A single non-zero entry $D_{in} = -1$ in column n of D indicates that the cutting pattern n for the second stage is cut from the intermediate roll i. Constraints (3.3) ensure that the demand for finished rolls given by the vector b is satisfied, and constraints (3.4) impose that the consumption of the intermediate rolls does not exceed their production. The objective is to minimize the total number of stock rolls required. Clearly, this problem is a special case of the generic model (MP), where A, a, d, and r are zero, and c is a vector of all ones. In general, there may be exponentially many feasible cutting patterns in both stages, which prompts us to develop a column generation algorithm for solving this formulation. The challenging issue is that each generated cutting pattern for the first stage, which includes an intermediate roll currently absent from the RMP, adds one more constraint to the model. Thus, the RMP grows both horizontally and vertically and exhibits the structure of a CDR-problem. MSCS satisfies Assumption 3.1.1 because a cutting pattern for the second stage based on an intermediate roll i cannot be generated unless there exists at least one cutting pattern for the first stage that includes this intermediate roll i. Moreover, the associated linking constraint is redundant in this case as required by Assumption 3.1.2, and any cutting pattern for the first stage that contains a currently absent intermediate roll i constitutes a minimal variable set for the corresponding linking constraint. The last assumption does also hold because the linking constraint corresponding to a currently absent intermediate roll is of the form (3.1). We conclude that MSCS belongs to the class of CDR-problems with no interaction.

In the column-and-row generation algorithm given in [73], three types of PSPs are defined. The first PSP looks for a new first-stage cutting pattern, which only includes the intermediate rolls that are already present in the restricted master problem. In the second PSP, the objective is to identify new cutting patterns for the second stage based on the currently existing intermediate rolls. Both of these PSPs are classical knapsack problems. The final PSP considers the possibility of generating both new intermediate rolls and related cutting patterns simultaneously and results in a difficult nonlinear integer programming problem. This subproblem is solved heuristically under a restrictive assumption which dictates that only one new intermediate roll can be generated at each iteration. Thus, the solution method given in [73] may terminate prematurely at a suboptimal solution which is verified by applying our proposed solution method to an instance provided in [72]. Our proposed solution method will be applied to MSCS in Section 4.2.1.

In the QSC problem, the objective is to cover all items $j \in J$ by the sets $k \in K$ at minimum total cost. In addition to the sum of the individual costs of the sets, we also incorporate a cross-effect between each pair of sets $k, l \in K$ which results in a quadratic objective function. [9] and [66] study this problem. QSC is formulated as

$$\begin{aligned} \text{minimize} \quad & y^{\intercal} F y, \\ \text{subject to} \quad & Ay \geq 1, \\ & y \in \{0,1\}^{|K|}, \end{aligned}$$

where A is a binary $\mid J \mid \times \mid K \mid$ matrix of set memberships, and F is a symmetric positive semidefinite $\mid K \mid \times \mid K \mid$ cost matrix. To linearize the objective function, we add a binary variable x_{kl} for each pair of sets $k, l \in K$. A set of linking constraints mandates that $x_{kl} = 1$ if and only if $y_k = y_l = 1$. Relaxing the integrality restrictions leads to the following linear program:

$$\begin{aligned} \text{minimize} \quad & \sum_{k \in K} f_{kk} y_k + \sum_{(k,l) \in P, k<l} 2 f_{kl} x_{kl}, && && (3.6) \\ \text{subject to} \quad & \sum_{k \in K} A_{jk} y_k \geq 1, && j \in J, && (3.7) \\ & y_k + y_l - x_{kl} \leq 1, && (k,l) \in P, \ k<l, && (3.8) \\ & y_k - x_{kl} \geq 0, && (k,l) \in P, \ k<l, && (3.9) \\ & y_l - x_{kl} \geq 0, && (k,l) \in P, \ k<l, && (3.10) \\ & y_k \geq 0, && k \in K, && (3.11) \\ & x_{kl} \geq 0, && (k,l) \in P, \ k<l, && (3.12) \end{aligned}$$

where $P := K \times K$ is the set of all possible pairs, and $A_{jk} = 1$, if item j is covered by set k; and 0, otherwise. The first set of constraints is the coverage constraints and the remaining are the linking constraints. This problem is a special case of the generic model (MP) with both B and b equal to zero, and a is a vector of ones. The vector of cost coefficients c and d in (MP) are formed by the diagonal and off-diagonal entries of the cost matrix F, respectively. To solve this formulation by column generation, we select a subset of the columns from K and the associated linking constraints to form the initial SRMP. If a new variable, say y_k, enters SRMP, a set of linking constraints and $x-$variables for each pair (k, l) with $l \in \bar{K}$ are also added. We note that the variable x_{kl} and the set of linking constraints $y_k + y_l - x_{kl} \leq 1$, $y_k - x_{kl} \geq 0$, and $y_l - x_{kl} \geq 0$ are redundant until both of the variables y_k and y_l are part of the SRMP. Thus, the minimal variable set $\{y_k, y_l\}$ allows us to generate x_{kl} and the constraints that relate these three variables. We arrive at the conclusion that QSC is a CDR-problem with interaction that satisfies both Assumptions 3.1.1 and 3.1.2 stipulated previously. Moreover, the set of linking constraints induced by any minimal variable set $S_K = \{y_k, y_l\}$ conforms to the characterization in Assumption 3.1.3 because the constraints (3.9) and (3.10) are of the form (3.1). In Section 4.2.2, we show that our proposed solution method for CDR-problems can handle the formulation (3.6)-(3.12).

For some problems, the linking constraints (3.8)-(3.10) may be formed by a strict subset $\bar{P}$ of the set of all possible pairs P. If in addition an explicit complete description of $\bar{P}$ is not available a priori before invoking a column generation algorithm, then we refer to these problems as *QSC with restricted pairs* (see also the discussion in the paragraph immediately following the statement of problem (MP).) Typically, in QSC problems with restricted pairs the generation of the pairs that belong to $\bar{P}$ requires a call to an oracle. One example is studied in [61] discussed in the next section.

Chapter 4

SIMULTANEOUS COLUMN-AND-ROW GENERATION

In this chapter, we develop a generic column-and-row generation algorithm that can handle all CDR-problems including our prototype examples QSC and MSCS as well as those mentioned in Section 2.3. First, we discuss the rationale of the proposed algorithm at a higher level without going into the details of the specific PSPs, and then analyze each type of subproblem separately. We devote most of the discussion to the row-generating PSP and to the proof of optimality of the proposed algorithm. Finally, the proposed algorithm is illustrated on three problems along with some computational experiments.

4.1 Proposed Solution Method

The dual of (MP) is given by

$$
\begin{aligned}
(\mathbf{DMP})\text{maximize} \quad & \sum_{j \in J} a_j u_j + \sum_{m \in M} b_m v_m + \sum_{i \in I} r_i w_i, \\
\text{subject to} \quad & \sum_{j \in J} A_{jk} u_j + \sum_{i \in I} C_{ik} w_i \leq c_k, \quad k \in K, \quad \text{(DMP-y)} \\
& \sum_{m \in M} B_{mn} v_m + \sum_{i \in I} D_{in} w_i \leq d_n, \quad n \in N, \quad \text{(DMP-x)} \\
& u_j \geq 0, j \in J, \quad v_m \geq 0, m \in M, \quad w_i \geq 0, i \in I,
\end{aligned}
$$

where u, v, and w denote the dual variables associated with the sets of constraints (MP-y), (MP-x), and (MP-yx), respectively.

As discussed in Chapter 1 and Chapter 2, the traditional column generation framework operates under the assumption that the number of constraints in the restricted master problem stays constant throughout the algorithm and all corresponding dual variables are known explicitly. This property is violated for CDR-problems, where generated columns introduce new constraints into the SRMP, and we need a new set of tools to solve these problems by column generation. In Section 1.1, we argued that the constraints missing in the SRMP may lead to a premature termination, if classical column generation is applied to the SRMP of a CDR-problem naively. To motivate our solution method and demonstrate our point formally, consider a set of variables $\{y_k | k \in S_K\}$, currently not present in the SRMP, and assume that adding these variables to the SRMP would also require adding a set of constraints $\Delta(S_K)$. Based on (DMP-y), the reduced cost $\bar{c}_k$ of $y_k, k \in S_K$, is then given by

$$\bar{c}_k = c_k - \sum_{j \in J} A_{jk} u_j - \sum_{i \in I(\bar{K}, \bar{N})} C_{ik} w_i - \sum_{i \in \Delta(S_K)} C_{ik} w_i, \tag{4.1}$$

and ignoring the dual variables $\{w_i | i \in \Delta(S_K)\}$ could result in

$$\bar{c}_k < 0 \leq c_k - \sum_{j \in J} A_{jk} u_j - \sum_{i \in I(\bar{K}, \bar{N})} C_{ik} w_i. \tag{4.2}$$

In this case, we fail to detect that y_k prices out favorably. In [2], such an error is committed as discussed in depth in [60].

Figure 4.1: The flow of the proposed column-and-row-generation algorithm.

An overview of the proposed column-and-row generation algorithm is depicted in Figure 4.1. The $y-$ and $x-$PSPs search for new $y-$ and $x-$ variables, respectively, under the assumption that these variables price out favorably with respect to the *current* set of rows in the SRMP. On the other hand, the row-generating PSP identifies at least one $y-$variable with a negative reduced cost, only if a set of new linking constraints and related $x-$variables are added to the SRMP. We note that not all CDR-problems give rise to all three PSPs as we discuss separately in the context of each PSP in the sequel. Theoretically, the order of invoking these subproblems does not matter; however, solving the row-generating PSP turns out to be computationally the most expensive in general. Therefore, we adopt the convention illustrated in Figure 4.1. The algorithm commences by calling the $y-$PSP repeatedly as long as new $y-$variables are generated, and then invokes the $x-$PSP in a similar manner. Finally, the row-generating PSP is called, if we can no longer generate $y-$ or $x-$ variables given the current set of constraints in the SRMP. Observe that we return to the $y-$PSP after solving a series of $x-$ or row-generating PSPs because the dual variables in the $y-$PSP are modified. The proposed column-and-row generation algorithm terminates, if solving the $y-$, $x-$, and the row-generating PSPs *consecutively in a single pass* does not yield a negatively priced column (only when FLAG=0 in Figure 4.1). Next, we investigate each PSP in detail.

4.1.1 $y-$Pricing Subproblem

This subproblem checks the feasibility of the dual constraints (DMP-y) using the values of the known dual variables. The objective is to determine a variable y_k, $k \in (K \setminus \bar{K})$ with a negative reduced cost. The $y-$PSP is stated as

$$\zeta_y = \min_{k \in (K \setminus \bar{K})} \quad \{c_k - \textstyle\sum_{j \in J} A_{jk} u_j - \sum_{i \in I(\bar{K}, \bar{N})} C_{ik} w_i\}, \tag{4.3}$$

where the dual variables $\{u_j | j \in J\}$ and $\{w_i | i \in I(\bar{K}, \bar{N})\}$ are obtained from the optimal solution of the current SRMP. If ζ_y is nonnegative, we move to the next subproblem. Otherwise, there exists y_k with $\bar{c}_k < 0$, and SRMP grows by a single variable by setting $\bar{K} \leftarrow \bar{K} \cup \{k\}$. For example, a column-and-row generation algorithm for the problems MSCS and QSC with restricted pairs requires this PSP.

At this point we note that whenever a column y_k with a negative reduced cost is generated, one or several minimal variable sets may be coincidentally completed by the introduction of this new variable. Consequently, it may become necessary, particularly for CDR-problems with interaction, to add the associated sets of linking constraints as well as the $x-$variables to the SRMP before re-invoking the $y-$PSP. For MSCS, this subproblem generates a cutting pattern for the first stage composed of the existing intermediate rolls only. Hence, no new linking constraint can be added. However, consider the QSC problem with restricted pairs and a pair of columns y_k and y_l, where $(k, l) \in \bar{P}$. When the $y-$PSP generates y_k, the associated column y_l may already be present in the SRMP. This would then require augmenting the problem with new constraints of type (3.8)-(3.10). Ultimately, when the $y-$PSP is unable to produce any more new columns, it is guaranteed that all linking constraints, which are induced by the minimal variable sets that are currently in the SRMP, are already generated. Although the $y-$PSP may yield new sets of linking constraints, we stress that it differs fundamentally from the row-generating PSP. In the former case, new linking constraints are only a by-product of the newly generated columns. However, the latter problem is solved with the sole purpose of identifying new linking constraints that help us price out additional $y-$variables which otherwise possess nonnegative reduced costs.

4.1.2 $x-$Pricing Subproblem

This subproblem attempts to generate a new $x-$variable by identifying a violated constraint (DMP-x) and assumes that the number of constraints in the SRMP is fixed. Recall from our previous discussion that no new linking constraint may be induced in the SRMP without generating new $y-$variables in the proposed column-and-row generation algorithm; that is, $\Delta(\emptyset) = \emptyset$ for this PSP (see also Assumption 3.1.1). Thus, all dual variables that appear in this PSP are known explicitly. The $x-$PSP is then simply given by

$$\zeta_x = \min_{n \in N_{\bar{K}}} \quad \{d_n - \textstyle\sum_{m \in M} B_{mn} v_m - \textstyle\sum_{i \in I(\bar{K}, \bar{N})} D_{in} w_i\}, \tag{4.4}$$

where the dual variables $\{v_m | m \in M\}$ and $\{w_i | i \in I(\bar{K}, \bar{N})\}$ are retrieved from the optimal solution of the current SRMP. In order to introduce a new variable x_n into the

SRMP, we require that at least one associated minimal set of variables $\{y_k|k \in S_K\}$ is already present in the model; that is, $S_K \subseteq \bar{K}$. Consequently, the search for x_n with a negative reduced cost in this PSP is restricted to the set $N_{\bar{K}} \subseteq N$, where $N_{\bar{K}}$ is the index set of all $x-$variables that may be induced by the set of variables $\{y_k|k \in \bar{K}\}$ in the current SRMP. We update $\bar{N} \leftarrow \bar{N} \cup \{n\}$ if $\zeta_x < 0$, i.e., if the $x-$PSP determines a variable $x_n, n \in N_{\bar{K}}$ that prices out favorably. Otherwise, the column-and-row generation algorithm continues with the appropriate subproblem dictated by the flow of the algorithm in Figure 4.1. In the MSCS problem, the $x-$PSP identifies cutting patterns for the second stage that only consume intermediate rolls that are produced by the cutting patterns for the first stage in the current SRMP. This PSP is not needed in a column-and-row generation algorithm for QSC-type problems because the $x-$variables in the corresponding formulations are auxiliary and are only added to the SRMP along with a set of new linking constraints induced by a set of new $y-$variables.

4.1.3 Row-Generating Pricing Subproblem

Note that before invoking the row-generating PSP, we always ensure that no negatively priced variables exist with respect to the current set of constraints in the SRMP (see Figure 4.1). Therefore, the objective of this PSP is to identify new columns that price out favorably *only after* adding new linking constraints currently absent from the SRMP. The primary challenge here is to properly account for the values of the dual variables of the missing constraints, and thus be able to determine which linking constraints should be added to the SRMP together with a set of variables. Demonstrating that this task can be accomplished implicitly is a fundamental contribution of the proposed solution framework. Under the assumptions for CDR-problems stated in Section 3.1, we can correctly anticipate the *optimal* values of the dual variables of the missing constraints without actually introducing them into the SRMP first, and this thinking-ahead approach enables us to calculate all reduced costs correctly in our column-and-row generation algorithm for CDR-problems. Furthermore, recall that Assumption 3.1.3 stipulates that a variable x_n that appears in a new linking constraint cannot assume a positive value unless all $y-$variables in an associated minimal variable set are positive. Thus, while we generate $x-$ and $y-$variables simultaneously in this PSP along with a set of linking constraints, the ultimate goal is to generate at least one $y-$variable with a negative reduced cost. We formalize these concepts later in the discussion.

In the context of the row-generating PSP, we need to distinguish between CDR-problems with and with no interaction as specified in Definition 3.1.1. For CDR-problems with no interaction, a single variable $y_k, k \notin \bar{K}$, may induce one or several new linking constraints. For instance, in the MSCS problem a cutting pattern $y_k, k \notin \bar{K}$, for the first stage leads to one new linking constraint per intermediate roll that it includes and is currently missing in the SRMP. Thus, all linking constraints that are required in the SRMP to decrease the reduced cost of y_k below zero may be directly induced by adding y_k to the SRMP. However, in CDR-problems with interaction no single variable y_k induces a set of new linking constraints, and the row-generating PSP must be capable of identifying one or several minimal variable sets, each with a cardinality larger than one, to add to the SRMP so that y_k prices out favorably in the presence of these one or several new sets of linking constraints. To illustrate this point for QSC, assume that the reduced cost of $y_k, k \notin \bar{K}$, is positive if we only consider the minimal variable sets of the form $\{y_k, y_l\}, l \in \bar{K}$. However, the reduced cost of y_k may turn negative if it is generated along with $y_{l'}, l' \notin \bar{K}$. In this case, $\{y_k, y_{l'}\}$ is a separate minimal variable set that introduces a set of linking constraints of the form (3.8)-(3.10) into the SRMP. Summarizing, the optimal solution of the row-generating PSP is a family $\mathcal{F}_k$ of index sets S_K^k, where each element $S_K^k \in \mathcal{F}_k$ is associated with a minimal variable set $\{y_l | l \in S_K^k\}$, and k in the superscript of the index set S_K^k denotes that $y_k \in \{y_l | l \in S_K^k\}$. Consequently, $\mathcal{F}_k$ is an element of the power set $\mathcal{P}_k$ of the set composed by the index sets of the minimal variable sets containing y_k. If the reduced cost $\bar{c}_k$ corresponding to the optimal family $\mathcal{F}_k$ is negative, then SRMP grows both horizontally and vertically with the addition of the variables $\{y_l | l \in \Sigma_k\}$, $\{x_n | n \in S_N(\Sigma_k)\}$, and the set of linking constraints $\Delta(\Sigma_k)$, where $\Sigma_k = \cup_{S_K^k \in \mathcal{F}_k} S_K^k$ denotes the index set of all $y-$variables introduced to the SRMP along with y_k. In the following discussion, SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ refers to the current SRMP formed by $\{y_k | k \in \bar{K}\}$,

$\{x_n | n \in \bar{N}\}$, and the set of linking constraints $I(\bar{K}, \bar{N})$ in addition to the structural constraints (SRMP-y)-(SRMP-x). Consequently, the outcome of the row-generating PSP is represented as SRMP$(\bar{K} \cup \Sigma_k, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$. In Table 4.1, we summarize our notation required for a detailed analysis of the row-generating PSP in the sequel.

A further distinction between CDR-problems with and with no interaction needs to be clarified before we delve into the mechanics of the row-generating PSP. The oracle that solves the row-generating PSP yields a family $\mathcal{F}_k$ –along with an associated index set Σ_k– so that $\bar{c}_k < 0$, if SRMP grows as specified above. For CDR-problems with no interaction, the optimal family of index sets reduces to a singleton, i.e., $\mathcal{F}_k = \{\{k\}\}$ and $\Sigma_k = \{k\}$. Furthermore, we must have $k \notin \bar{K}$; otherwise, $\bar{c}_k \geq 0$ would hold because $k \in \bar{K}$ implies $S_N(\{k\}) \subseteq \bar{N}$ and $\Delta(\{k\}) \subseteq I(\bar{K}, \bar{N})$, and the current SRMP would have been solved to optimality with all constraints relevant for y_k. On the other hand, for CDR-problems with interaction there may exist an $l \in \Sigma_k$ with $l \in \bar{K}$.

Table 4.1: Notation for the analysis of the row-generating PSP.

S_K	denotes the index set of a minimal variable set $\{y_l \vert l \in S_K\}$.
S_K^k	denotes that y_k is a member of the minimal variable set $\{y_l \vert l \in S_K^k\}$.
$S_N(S_K)$	denotes the index set of the $x-$variables induced by $\{y_l \vert l \in S_K\}$.
$\Delta(S_K)$	denotes the index set of the linking constraints induced by $\{y_l \vert l \in S_K\}$.
$\mathcal{P}_k$	denotes the power set of the set composed by the index sets of the minimal variable sets containing y_k.
$\mathcal{F}_k$	denotes a family of the index sets of the minimal variable sets of the form S_K^k, i.e., $\mathcal{F}_k \in \mathcal{P}_k$.
Σ_k	$= \cup_{S_K^k \in \mathcal{F}_k} S_K^k$.
SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$	denotes the current SRMP formed by $\{y_k \vert k \in \bar{K}\}$, $\{x_n \vert n \in \bar{N}\}$, and the set of linking constraints $I(\bar{K}, \bar{N})$ in addition to (SRMP-y)-(SRMP-x).

As explained previously, the minimal variable set $\{y_l | l \in S_K\}$ introduces $\Delta(S_K)$. In general, this relationship is not one-to-one; that is, y_k may appear in several sets of linking constraints, and the same set of linking constraints may be induced by several different minimal variable sets. To illustrate in the context of MSCS, if the intermediate rolls i, j and i, h appear in the first-stage cutting patterns k and l, respectively, then we have $\{(\{k\}, \{i\}), (\{k\}, \{j\}), (\{l\}, \{i\}), (\{l\}, \{h\})\}$, where a pair $(S_K, \Delta(S_K))$ specifies that the minimal variable set $\{y_l | l \in S_K\}$ introduces $\Delta(S_K)$. Therefore, $\{y_k\}$ and $\{y_l\}$ are the minimal variable sets for the sets of linking constraints $\{i, j\}$ and $\{i, h\}$, respectively, and the linking constraint i may be induced by both $\{y_k\}$ and $\{y_l\}$. In contrast, for the QSC problem, each set of linking constraints of the form (3.8)-(3.10) is introduced to the SRMP by a unique minimal variable set $\{y_k, y_l\}$, and we have $(\{k, l\}, \{i_1, i_2, i_3\})$, where i_1, i_2, i_3, are the indices of the associated linking constraints.

In general, adding new constraints and variables to an LP may destroy both the primal and the dual feasibility. In our case, Assumption 3.1.2 guarantees that the primal feasibility is preserved. Therefore, the goal of our analysis is to attach a correct set of values to each variable $w_i, i \in \Delta(\Sigma_k)$, and thus be able to calculate the reduced costs of y_k and $\{x_n | n \in S_N(\Sigma_k)\}$ to be inserted into the SRMP correctly. In particular, the ensuing analysis computes the optimal values of $\{w_i | i \in \Delta(\Sigma_k)\}$ without solving the SRMP explicitly under the presence of the currently missing associated set of linking constraints $\Delta(\Sigma_k)$. Moreover, it also guarantees that the optimal values of the dual variables $\{u_j | j \in J\}$, $\{v_m | m \in M\}$, and $\{w_i | i \in I(\bar{K}, \bar{N})\}$ retrieved from the optimal solution of the current SRMP would remain optimal with respect to the SRMP augmented with the set of linking constraints $\Delta(\Sigma_k)$ and $\{x_n | S_N(\Sigma_k)\}$. These properties, stated formally in Corollary 4.1.1a-b, are key to the correctness of the proposed column-and-row generation algorithm. Then, for any given y_k, an associated $\mathcal{F}_k$, and $S_K^k \in \mathcal{F}_k$, we have

$$\bar{c}_k = c_k - \sum_{j \in J} A_{jk} u_j - \sum_{i \in I(\bar{K}, \bar{N})} C_{ik} w_i - \sum_{i \in \Delta(\Sigma_k)} C_{ik} w_i, \tag{4.5}$$

$$\bar{d}_n = d_n - \sum_{m \in M} B_{mn} v_m - \sum_{i \in I(\bar{K}, \bar{N})} D_{in} w_i - \sum_{i \in \Delta(\Sigma_k)} D_{in} w_i \tag{4.6}$$

$$= d_n - \sum_{m \in M} B_{mn} v_m - \sum_{i \in \Delta(S_K^k)} D_{in} w_i, \tag{4.7}$$

where $\bar{c}_k$ and $\bar{d}_n$ are the reduced costs for y_k and $x_n, n \in S_N(S_K^k)$, respectively. The simplification of expression (4.6) to (4.7) follows from Assumption 3.1.1 which states that an $x-$variable appears in no more than one set of linking constraints. To reiterate, in (4.5)-(4.7) the values of the dual variables $\{u_j|j \in J\}$, $\{v_m|m \in M\}$, and $\{w_i|i \in I(\bar{K}, \bar{N})\}$ are retrieved from the optimal solution of the current SRMP, and $\{w_i|i \in \Delta(\Sigma_k)\}$ are unknown. Next, we introduce a series of conditions imposed on the reduced costs (4.7) as well as on the unknown dual variables $\{w_i|i \in \Delta(\Sigma_k)\}$ which ultimately leads to the formulation of the row-generating PSP. In this discussion, we also present how we can obtain a valid starting basis for the next optimization of the SRMP given that it is augmented by the variables $\{y_l|l \in \Sigma_k\}$, $\{x_n|n \in S_N(\Sigma_k)\}$, and the set of linking constraints $\Delta(\Sigma_k)$.

Suppose that for a given $\mathcal{F}_k$ and a set of associated dual variables $\{w_i|i \in \Delta(\Sigma_k)\}$ we have $\bar{c}_k \geq 0$ and $\bar{d}_{n'} < 0$ for some $n' \in S_N(S_K^k)$ with $S_K^k \in \mathcal{F}_k$. Hence, $\{y_l|l \in \Sigma_k\}$, $\{x_n|n \in S_N(\Sigma_k)\}$, and the set of linking constraints $\Delta(\Sigma_k)$ are added to the SRMP. This implies that $x_{n'}$ is eligible to enter the basis during the next iteration of solving the SRMP. However, this basis update would only result in a degenerate simplex iteration as the value of $x_{n'}$ is forced to zero in the basis by Assumption 3.1.3. That is, there exists a nonbasic variable y_l, $l \in S_K^k$, such that an associated constraint (3.1) is introduced into the SRMP. Note that the existence of such a nonbasic variable is guaranteed because $S_K^k \not\subseteq \bar{K}$. In order to avoid this type of degeneracy, we require that $\bar{d}_n \geq 0$ holds for all $n \in S_N(S_K^k)$ and for each $S_K^k \in \mathcal{F}_k$ while determining the values of $\{w_i|i \in \Delta(\Sigma_k)\}$. In other words, we impose the following set of constraints:

$$\sum_{m \in M} B_{mn} v_m + \sum_{i \in \Delta(S_K^k)} D_{in} w_i \leq d_n, \quad n \in S_N(S_K^k), S_K^k \in \mathcal{F}_k. \tag{4.8}$$

We underline that our proposed approach goes beyond the classical LP sensitivity analysis that would augment the basis with the surplus variables in the new linking constraints and then proceed to repair the infeasibility in the constraints (4.8). This is because setting $w_i = 0$, $i \in S_K^k$, $S_K^k \in \mathcal{F}_k$ may violate (4.8). Therefore, incorporating these constraints directly into the row-generating PSP may be regarded as a look-ahead feature. A further critical observation is that constraints (4.8) exhibit a block-diagonal structure. Given the optimal solution of the current SRMP, the first term on the left hand side of (4.8) is a constant for all n, and hence, we have

$$\sum_{i \in \Delta(S_K^k)} D_{in} w_i \leq d_n - \sum_{m \in M} B_{mn} v_m, \quad n \in S_N(S_K^k), S_K^k \in \mathcal{F}_k, \tag{4.9}$$

which exposes the block-diagonal structure. The dual variables $\{w_i | i \in \Delta(S_K^k)\}$ do not factor into the reduced costs of any $x-$variables, except for $\{x_n | n \in S_N(S_K^k)\}$. Thus, the task of determining the values of $\{w_i | i \in \Delta(\Sigma_k)\}$ decomposes, and this property is also exploited in our analysis. We next show that enforcing the set of constraints (4.8) in the row-generating PSP does not change the *minimum value* of $\bar{c}_k$ and hence imposing (4.8) does not affect the correctness of the column-and-row generation procedure.

LEMMA 4.1.1 *For a given k, an associated $\mathcal{F}_k$, and $S_K^k \in \mathcal{F}_k$, imposing (4.8) on the set of unknown dual variables $\{w_i | i \in \Delta(S_K^k)\}$ while solving the row-generating PSP does not increase the minimum value of $\bar{c}_k$.*

PROOF. This result stems directly from Assumption 3.1.3 which states that there always exists a linking constraint $i' \in \Delta(S_K^k)$ of the form (3.1) such that $C_{i'k} > 0$ and $D_{i'n} < 0$ for all $n \in S_N(S_K^k)$. Coupling this with $w_i \geq 0, i \in \Delta(S_K^k)$ as required by the (DMP), we conclude that increasing $w_{i'}$ increases the reduced cost $\bar{d}_n$ given in (4.7) for all $\{x_n | n \in S_N(S_K^k)\}$ while reducing $\bar{c}_k$ in (4.5). Thus, (4.8) is always satisfied for the minimum value of $\bar{c}_k$. □

From the discussion so far it is evident that the row-generating PSP must provide us with a variable y_k and an associated family of index sets $\mathcal{F}_k$ so that the reduced cost $\bar{c}_k$ as defined in (4.5) is negative. Thus, for a given variable y_k we need to select a subset $\mathcal{F}_k \in \mathcal{P}_k$ so that $\bar{c}_k$ is minimized. During this optimization we must prescribe that the values determined for the unknown set of dual variables $\{w_i | i \in \Delta(S_K^k)\}$ satisfy the conditions set forth in (4.8) for each $S_K^k \in \mathcal{F}_k$. These arguments prompt us to pose the row-generating PSP as a two-stage optimization problem. In the first stage, we formulate and solve the problem of finding the minimum reduced cost for a given y_k as a subset selection problem. For any given $\mathcal{F}_k \in \mathcal{P}_k$, the problem of computing the optimal values of $\{w_i | i \in \Delta(\Sigma_k)\}$ decomposes into finding the optimal values of

$\{w_i | i \in \Delta(S_K^k)\}$ for each $S_K^k \in \mathcal{F}_k$. In the second stage, we pick the $y-$variable with the most negative minimum reduced cost. We stop solving the row-generating PSP and proceed according to Figure 4.1 if the minimum reduced cost is nonnegative for all $y_k, k \in (K \setminus \bar{K})$.

The only missing piece in the approach described in the preceding paragraph is computing a valid reduced cost for y_k for a given $\mathcal{F}_k$ without changing the reduced costs of the variables in SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$. This task is accomplished by showing that the optimal solution of the row-generating PSP corresponds to an *implicit construction of a basic optimal solution* to SRMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$ that allows us to correctly price out $\{y_l | l \in \Sigma_k\}$. In particular, we prove that the optimal values of the dual variables $\{u_j | j \in J\}$, $\{v_m | m \in M\}$, and $\{w_i | i \in I(\bar{K}, \bar{N})\}$ in SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ are identical to those in SRMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$, and the values set for $\{w_i | i \in \Delta(\Sigma_k)\}$ in the row-generating PSP are optimal for SRMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$ as stated in Corollary 4.1.1a-b. In addition, it turns out that we have $C_{il} w_i = 0$ for a variable $y_l, l \in \bar{K}$ and for $i \in \Delta(\Sigma_k)$. In other words, a $y-$variable that currently exists in the SRMP does not appear in a new linking constraint with a positive dual variable, and this property guarantees that the reduced costs of $\{y_l | l \in \bar{K}\}$ are identical with respect to the optimal dual solutions of both SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ and SRMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$ as stated in Corollary 4.1.1c.

To explain the construction of an optimal basis for SRMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$ based on the solution of the row-generating PSP, suppose that we are given a specific $\mathcal{F}_k$. We introduce $\{x_n | n \in S_N(\Sigma_k)\}$ and a set of new linking constraints $\Delta(\Sigma_k)$ into SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ to obtain SRMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$. Warm starting the primal simplex method for this new SRMP would require us to augment the optimal basis of SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ with $| \Delta(\Sigma_k) |$ new basic variables associated with the new set of linking constraints. To ensure complementary slackness, we determine the values of $\{w_i | i \in \Delta(\Sigma_k)\}$ such that the number of linearly independent active constraints among $w_i \geq 0, i \in \Delta(\Sigma_k)$ and (4.9) is at least $| \Delta(\Sigma_k) |$. This restriction is directly added to the definition of the row-generating PSP specified below. A tight constraint of the form (4.9) prescribes adding the corresponding $x-$variable to the basis, while $w_i = 0$ implies that the basis is extended by the corresponding primal surplus variable. In addition, in order to ensure that $C_{il} w_i = 0$ for $i \in \Delta(\Sigma_k)$ and for $\{y_l | l \in \bar{K}\}$ as discussed before, we only allow $w_i > 0$ if constraint $i \in \Delta(S_K^k)$ is of the form (3.1) as specified in Assumption 3.1.3 with $C_{ik} > 0$. Clearly, such a constraint does not include a variable $y_l, l \in \bar{K}$. The index set of constraints $\Delta(S_K^k)$ of the form (3.1) with $C_{ik} > 0$ is represented by $\Delta_+(S_K^k)$, and the complement of this set is denoted by $\Delta_0(S_K^k) = \Delta(S_K^k) \setminus \Delta_+(S_K^k)$. Thus, we always pick a surplus variable as basic for a constraint $i \in \Delta_0(S_K^k)$ for all $S_K^k \in \mathcal{F}_k$. For the other new linking constraints, we either designate an $x-$ or a surplus variable as basic. In Lemma 4.1.3, we first prove that the augmentation prescribed by the row-generating PSP is a valid basis for $\text{SRMP}(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$, and then in Lemma 4.1.4, we prove that it is optimal. In particular, the values of the dual variables $\{w_i | i \in \Delta(\Sigma_k)\}$ set as described turn out to be optimal for $\text{SRMP}(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$ as formalized in Corollary 4.1.1b. The row-generating PSP is then stated as:

$$\zeta_{yx} = \min_{k \in (K \setminus \bar{K})} \left\{ c_k - \sum_{j \in J} A_{jk} u_j - \sum_{i \in I(\bar{K}, \bar{N})} C_{ik} w_i - \max_{\mathcal{F}_k \in \mathcal{P}_k} \left(\sum_{S_K^k \in \mathcal{F}_k} \alpha_{S_K^k} \right) \right\}, \text{ where} \tag{4.10}$$

$$\begin{aligned}
\alpha_{S_K^k} = \text{maximize} \quad & \sum_{i \in \Delta(S_K^k)} C_{ik} w_i, && \text{(4.11a)}\\
\text{subject to} \quad & \sum_{i \in \Delta(S_K^k)} D_{in} w_i \leq d_n - \sum_{m \in M} B_{mn} v_m, \quad n \in S_N(S_K^k), && \text{(4.11b)}\\
& w_i = 0, \quad i \in \Delta_0(S_K^k), && \text{(4.11c)}\\
& w_i \geq 0, \quad i \in \Delta_+(S_K^k), && \text{(4.11d)}\\
& |\Delta(S_K^k)| \text{ many linearly independent tight constraints among} \\
& \text{(4.11b)} - \text{(4.11d)}. && \text{(4.11e)}
\end{aligned}$$

The fundamental property of this formulation is that we solve (4.11) independently for each $S_K^k \in \mathcal{F}_k$ which allows us to calculate the minimum reduced cost of y_k efficiently. This decomposition relies on the block-diagonal structure previously discussed in the context of (4.9) and is exemplified in Section 4.2 when our generic methodology is applied to the MSCS and QSC problems. A potential source of difficulty is the constraint (4.11e) which mandates that the search for an optimal solution of (4.11) is restricted to the set of extreme points of the polyhedron described by (4.11b)-(4.11d). Without this restriction, the problem (4.11a)-(4.11d) is unbounded by a similar argument to that used in the proof of Lemma 4.1.1. Fortunately, in many cases (4.11) is amenable to simple solution approaches. This is illustrated on the MSCS and QSC problems in Section 4.2.

In summary, suppose that solving the row-generating PSP (4.10)-(4.11) results in $\bar{c}_k = \zeta_{yx} < 0$ and an associated family of index sets $\mathcal{F}_k$. Then, SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ expands to SRMP$(\bar{K} \cup \Sigma_k, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$ before the primal simplex method is warm started based on the basis augmentation provided by the optimal solutions of (4.11) for $S_K^k \in \mathcal{F}_k$. This augmentation achieves two primary goals. First, the resulting basis is optimal for SRMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$, and the optimal objective function value ζ_{yx} of the row-generating PSP is the correct reduced cost of y_k under this augmentation. Second, we can invoke the primal simplex algorithm with this initial basis for SRMP$(\bar{K} \cup \Sigma_k, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$ so that y_k is the natural candidate to enter the basis. In the remainder of this section, we prove these properties of the proposed basis augmentation preceding a formal proof of the correctness of the proposed column-and-row generation approach for CDR-problems.

Let $\mathbf{B}$ and $\mathcal{B}$ be the optimal basis of SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ and the associated basic sequence, respectively. Suppose that $\mathbf{B}$ is a $\beta \times \beta$ matrix, and $\delta := \mid \Delta(\Sigma_k) \mid$ denotes the number of new constraints to be added to the SRMP. Recall that we always pick surplus variables as basic for the set of constraints $\Delta_0(S_K^k)$ for all $S_K^k \in \mathcal{F}_k$. However, for a constraint $i \in \Delta_+(S_K^k)$ we either select the corresponding surplus variable as basic if $w_i = 0$ or an $x-$variable that appears in this constraint, if its associated dual constraint (4.11b) is tight in the optimal solution of (4.11) for S_K^k. In other words, no more than $\mid \Delta_+(S_K^k) \mid$ of the variables $\{x_n | n \in S_N(S_K^k)\}$ are designated as basic by the optimal solution of (4.11). We denote the sets of new linking constraints associated with the new basic $x-$ and surplus variables as $\Delta_x(\Sigma_k)$ and $\Delta_s(\Sigma_k)$, respectively, where $\delta_x = \mid \Delta_x(\Sigma_k) \mid$, $\delta_s = \mid \Delta_s(\Sigma_k) \mid$, and $\Delta_x(\Sigma_k) \subseteq \cup_{S_K^k \in \mathcal{F}_k} \Delta_+(S_K^k)$. The resulting augmented matrix $\mathbf{B}_k$ is then obtained as:

$$\mathbf{B}_k = \left(\begin{array}{ccc|cc} \mathbf{A_1} & \mathbf{0} & \mathbf{E_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B_1} & \mathbf{E_2} & \mathbf{B_2} & \mathbf{0} \\ \mathbf{C_1} & \mathbf{D_1} & \mathbf{E_3} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{D_2} & \mathbf{0} \\ \mathbf{C_2} & \mathbf{0} & \mathbf{0} & \mathbf{D_3} & -\mathbf{I} \end{array}\right) = \left(\begin{array}{c|cc} \mathbf{B} & \mathbf{F} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{D_2} & \mathbf{0} \\ \mathbf{G} & \mathbf{D_3} & -\mathbf{I} \end{array}\right), \tag{4.12}$$

where the coefficients of the new basic $x-$variables in the currently existing constraints in the SRMP are given by a $\beta \times \delta_x$ matrix $\mathbf{F} = \left(\begin{smallmatrix} \mathbf{0} \\ \mathbf{B_2} \\ \mathbf{0} \end{smallmatrix}\right)$. The $\delta_x \times \delta_x$ matrix $\mathbf{D_2}$ and the $\delta_s \times \delta_x$ matrix $\mathbf{D_3}$ specify the coefficients of these $x-$variables in the new linking constraints $\Delta_x(\Sigma_k)$ and $\Delta_s(\Sigma_k)$, respectively. The final column of $\mathbf{B}_k$ is associated with the new basic surplus variables, where $\mathbf{I}$ is a $\delta_s \times \delta_s$ identity matrix. The $\delta_x \times \beta$ matrix $\begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}$ in the fourth row of $\mathbf{B}_k$ and the $\delta_s \times \beta$ matrix $\mathbf{G} = \begin{pmatrix} \mathbf{C_2} & \mathbf{0} & \mathbf{0} \end{pmatrix}$ are constructed by the coefficients of the current basic variables in the new linking constraints $\Delta_x(\Sigma_k)$ and $\Delta_s(\Sigma_k)$, respectively. This partitioning is best explained in the context of the illustration in Figure 4.2 for the QSC problem, for which $\mathbf{B_1} = \mathbf{E_2} = \mathbf{B_2} = \mathbf{0}$ because the $x-$ variables appear only in the linking constraints. In this specific example, $\zeta_{yx} = \bar{c}_k < 0$ and $\mathcal{F}_k = \{\{k,l\},\{k,m\}\}$. The variable y_l is already present in the current SRMP, and y_m is to be incorporated in the SRMP along with y_k. Along with these, we introduce x_{kl}, x_{km}, two sets of linking constraints of the form (3.8)-(3.10) associated with the pairs of variables y_k, y_l, and y_k, y_m, respectively, and a set of six surplus variables associated with the new linking constraints into the SRMP. The problem (4.11) designates x_{kl}, s_{l1}, and s_{l3} as basic for the constraints $y_k - x_{kl} - s_{l2} = 0$, $-y_k - y_l + x_{kl} - s_{l1} = -1$, and $y_l - x_{kl} - s_{l3} = 0$, respectively, where the first of these constraints belongs to the set $\Delta_+(\{k,l\})$ and the rest form the set $\Delta_0(\{k,l\})$, respectively. Note that s_{l2} may replace x_{kl} in the augmented basis depending on the optimal solution of (4.11) for $\{k,l\}$. The variables x_{km}, s_{m1}, and s_{m3} are selected as basic for the set of linking constraints $\Delta(\{k,m\})$ in a similar way. Thus, $\Delta_x(\{k,l,m\})$ consists of the new linking constraints $y_k - x_{kl} - s_{l2} = 0$ and $y_k - x_{km} - s_{m2} = 0$, while the rest of the new linking constraints belong to $\Delta_s(\{k,l,m\})$. Two crucial observations are due based on this discussion. First, no variable in the current SRMP is present in a constraint $i \in \Delta_+(S_K^k)$ for any $S_K^k \in \mathcal{F}_k$; that is, the submatrix in the first position in the fourth row of $\mathbf{B}_k$ is zero. Second, $\mathbf{D_2}$ is invertible as formalized by the next lemma. These two properties allow us to establish that $\mathbf{B}_k$ is a valid basis for $\text{SRMP}(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$ in Lemma 4.1.3. For other CDR-problems with interaction, we would need to define the sets $\Delta_+(S_K^k)$ and $\Delta_0(S_K^k)$ as appropriate for all $S_K^k \in \mathcal{F}_k$, and the structure of the submatrices $\mathbf{C_2}$, $\mathbf{D_2}$, and $\mathbf{D_3}$ would be different. Otherwise, the basis augmentation carries over in exactly the same way. The only extra provision for CDR-problems with no interaction is that $\mathbf{C_2} = \mathbf{0}$ because $\Sigma_k = \{k\}$ and $k \notin \bar{K}$.

Constraint	Basic Variable	$SRMP(\bar{K}, \bar{N}, I(\bar{K}, \bar{N})$			$\mathbf{B_1}$	$\mathbf{E_2}$	$\mathbf{B_2}$			
		$y_j, j \in \mathcal{B}$ / y_l	$x_n, n \in \mathcal{B}$	basic surplus vars	x_{kl}	x_{km}	s_{l1}	s_{l3}	s_{m1}	s_{m3}
J		$\mathbf{A_1}$	$\mathbf{0}$	$\mathbf{E_1}$	$\mathbf{0}$				$\mathbf{0}$	
M	$\mathcal{B}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$				$\mathbf{0}$	
$I(\bar{K}, \bar{N})$		$\mathbf{C_1}$	$\mathbf{D_1}$	$\mathbf{E_3}$	$\mathbf{0}$				$\mathbf{0}$	
$\Delta_x(\{k,l,m\})$	x_{kl}	0	$\mathbf{0}$	$\mathbf{0}$	-1				$\mathbf{0}$	
	x_{km}					-1				
	s_{l1}	-1			1		-1			
$\Delta_s(\{k,l,m\})$	s_{l3}	1	$\mathbf{0}$	$\mathbf{0}$	-1			-1		
	s_{m1}					1			-1	
	s_{m3}					-1				-1
		$\mathbf{C_2}$		$\mathbf{D_2}$			$\mathbf{D_3}$		$\mathbf{-I}$	

Figure 4.2: Basis augmentation for QSC, where $\mathcal{F}_k = \{\{k,l\},\{k,m\}\}$, and the new basic variables $\{x_{kl}, s_{l1}, s_{l3}\}$ and $\{x_{km}, s_{m1}, s_{m3}\}$ are associated with the new linking constraints $\Delta(\{k,l\})$ and $\Delta(\{k,m\})$, respectively.

Lemma 4.1.2 *The $\delta_x \times \delta_x$ matrix $\mathbf{D_2}$ is invertible.*

Proof. The matrix $\begin{pmatrix} \mathbf{D_2} & \mathbf{0} \\ \mathbf{D_3} & \mathbf{-I} \end{pmatrix}$ is constructed by solving (4.11) for each $S_K^k \in \mathcal{F}_k$ and exhibits a block-diagonal structure as discussed before. The columns in a given block are linearly independent as prescribed by (4.11e). Therefore, $\begin{pmatrix} \mathbf{D_2} & \mathbf{0} \\ \mathbf{D_3} & \mathbf{-I} \end{pmatrix}$ must be invertible, and by the uniqueness of the inverse we conclude that $\begin{pmatrix} \mathbf{D_2} & \mathbf{0} \\ \mathbf{D_3} & \mathbf{-I} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{D_2}^{-1} & \mathbf{0} \\ \mathbf{D_3}\mathbf{D_2}^{-1} & \mathbf{-I} \end{pmatrix}$. Thus, $\mathbf{D_2}$ must be invertible. $\square$

In Figure 4.2, one block in $\begin{pmatrix} \mathbf{D_2} & \mathbf{0} \\ \mathbf{D_3} & \mathbf{-I} \end{pmatrix}$ is formed by the coefficients of x_{kl}, s_{l1}, and s_{l3}, while x_{km}, s_{m1}, and s_{m3} construct the second block. The next lemma proves that $\mathbf{B}_k$ provides us with a basic solution for SRMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$.

LEMMA 4.1.3 $\mathbf{B}_k$ *is a* $(\beta+\delta)$*-dimensional basis for* SRMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$*, and its inverse is obtained as:*

$$\begin{pmatrix} \mathbf{B} & \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{D_2} & \mathbf{0} \\ \mathbf{G} & \mathbf{D_3} & -\mathbf{I} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{B}^{-1} & -\mathbf{B}^{-1}\mathbf{F}\mathbf{D_2}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{D_2}^{-1} & \mathbf{0} \\ \mathbf{G}\mathbf{B}^{-1} & -\mathbf{G}\mathbf{B}^{-1}\mathbf{F}\mathbf{D_2}^{-1} + \mathbf{D_3}\mathbf{D_2}^{-1} & -\mathbf{I} \end{pmatrix}.$$

PROOF. The matrix $\mathbf{J} = \begin{pmatrix} \mathbf{B} & \mathbf{F} \\ \mathbf{0} & \mathbf{D_2} \end{pmatrix}$ is invertible because both $\mathbf{B}$ and $\mathbf{D_2}$ are invertible, and we compute $\mathbf{J}^{-1} = \begin{pmatrix} \mathbf{B}^{-1} & -\mathbf{B}^{-1}\mathbf{F}\mathbf{D_2}^{-1} \\ \mathbf{0} & \mathbf{D_2}^{-1} \end{pmatrix}$. Thus, $\mathbf{B}_k = \begin{pmatrix} \mathbf{J} & \mathbf{0} \\ \mathbf{K} & -\mathbf{I} \end{pmatrix}$, where $\mathbf{K} = \begin{pmatrix} \mathbf{G} & \mathbf{D_3} \end{pmatrix}$. Finally, we obtain

$$\mathbf{B}_k^{-1} = \begin{pmatrix} \mathbf{J} & \mathbf{0} \\ \mathbf{K} & -\mathbf{I} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{J}^{-1} & \mathbf{0} \\ \mathbf{K}\mathbf{J}^{-1} & -\mathbf{I} \end{pmatrix} = \begin{pmatrix} \mathbf{B}^{-1} & -\mathbf{B}^{-1}\mathbf{F}\mathbf{D_2}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{D_2}^{-1} & \mathbf{0} \\ \mathbf{G}\mathbf{B}^{-1} & -\mathbf{G}\mathbf{B}^{-1}\mathbf{F}\mathbf{D_2}^{-1} + \mathbf{D_3}\mathbf{D_2}^{-1} & -\mathbf{I} \end{pmatrix}$$

after plugging in $\mathbf{J}^{-1}$ and $\mathbf{K}\mathbf{J}^{-1}$ as appropriate. □

We next state one of our main results in this section and prove that $\mathbf{B}_k$ is in fact an optimal basis for SRMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$. We emphasize that this result does not require an optimal solution of (4.11) for $S_K^k \in \mathcal{F}_k$. It is sufficient to choose any extreme point feasible solution of (4.11b)-(4.11d) for each $S_K^k \in \mathcal{F}_k$ while constructing $\mathbf{B}_k$.

LEMMA 4.1.4 $\mathbf{B}_k$ *is an optimal basis for* SRMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$.

PROOF. It is sufficient to prove that $\mathbf{B}_k$ defines a pair of primal and dual basic feasible solutions since complementary slackness is always satisfied by a basic solution. In the following, $\mathbf{b}$ represents the right hand side coefficients of the constraints in SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ while $\mathbf{c}_{\mathcal{B}}$ stands for the objective function coefficients of the variables in the optimal basic sequence $\mathcal{B}$ of SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$. Furthermore, the objective coefficients of the new basic $x-$variables are denoted by $\mathbf{c}_x$, and the right hand sides of the new linking constraints $\Delta_x(\Sigma_k)$ and $\Delta_s(\Sigma_k)$ are given by the vectors $\mathbf{r}_x$ and $\mathbf{r}_s$, respectively, where $\mathbf{r}_x = \mathbf{0}$ by (3.1) in Assumption 3.1.3. Thus, the vector $\mathbf{b}_k = \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{r}_s \end{pmatrix}$ defines the right hand side of SRMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$.

For verifying the primal feasibility, we compute

$$\mathbf{B}_k^{-1}\mathbf{b}_k = \begin{pmatrix} \mathbf{B}^{-1} & -\mathbf{B}^{-1}\mathbf{F}\mathbf{D_2}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{D_2}^{-1} & \mathbf{0} \\ \mathbf{G}\mathbf{B}^{-1} & -\mathbf{G}\mathbf{B}^{-1}\mathbf{F}\mathbf{D_2}^{-1} + \mathbf{D_3}\mathbf{D_2}^{-1} & -\mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{b} \\ \mathbf{0} \\ \mathbf{r}_s \end{pmatrix} = \begin{pmatrix} \mathbf{B}^{-1}\mathbf{b} \\ \mathbf{0} \\ \mathbf{G}\mathbf{B}^{-1}\mathbf{b} - \mathbf{r}_s \end{pmatrix} \geq \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}. \quad (4.13)$$

The nonnegativity of $\mathbf{B}^{-1}\mathbf{b}$ follows from the optimality of $\mathbf{B}$ for SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$. By Assumption 3.1.2, the optimal solution of SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ does not violate the linking constraints $\Delta_s(\Sigma_k)$ which are absent from SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$. Thus, we have $\mathbf{G}(\mathbf{B}^{-1}\mathbf{b}) \geq \mathbf{r}_s$, and $\mathbf{G}\mathbf{B}^{-1}\mathbf{b} - \mathbf{r}_s \geq \mathbf{0}$ as required.

In order to check the nonnegativity of the reduced costs in SRMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k)$, $I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$, we first determine the values of the dual variables prescribed by $\mathbf{B}_k$. That is

$$\begin{aligned} &\begin{pmatrix} \mathbf{c}_{\mathcal{B}} & \mathbf{c}_x & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{B}^{-1} & -\mathbf{B}^{-1}\mathbf{F}\mathbf{D_2}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{D_2}^{-1} & \mathbf{0} \\ \mathbf{G}\mathbf{B}^{-1} & -\mathbf{G}\mathbf{B}^{-1}\mathbf{F}\mathbf{D_2}^{-1} + \mathbf{D_3}\mathbf{D_2}^{-1} & -\mathbf{I} \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{c}_{\mathcal{B}}\mathbf{B}^{-1} & (\mathbf{c}_x - \mathbf{c}_{\mathcal{B}}\mathbf{B}^{-1}\mathbf{F})\mathbf{D_2}^{-1} & \mathbf{0} \end{pmatrix}, \end{aligned} \quad (4.14)$$

where the objective coefficients of the basic surplus variables are represented by $\mathbf{0}$. From (4.14), we conclude that the values of the dual variables $\{u_j | j \in J\}$, $\{v_m | m \in M\}$, and $\{w_i | i \in I(\bar{K}, \bar{N})\}$ are identical to those in the optimal solution of SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$. Moreover, we can show that the values of the dual variables $\{w_i | i \in \Delta(\Sigma_k)\}$ are precisely those assigned by the row-generating PSP. To this end, recall that we form an invertible $\delta \times \delta$ submatrix $\begin{pmatrix} \mathbf{D_2} & \mathbf{0} \\ \mathbf{D_3} & -\mathbf{I} \end{pmatrix}$ consisting of columns corresponding to $x-$ and surplus variables based on the solutions of (4.11) for each $S_K^k \in \mathcal{F}_k$. In addition, note that the objective function coefficient of $x_n, n \in S_N(S_K)$ in the primal LP corresponding to the dual LP (4.11a)-(4.11d) is given by $d_n - \sum_{m \in M} B_{mn} v_m$. Clearly, if x_n is selected as basic in $\mathrm{SRMP}(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$ this value is equal to the component of $\mathbf{c}_x - \mathbf{c}_{\mathcal{B}}\mathbf{B}^{-1}\mathbf{F}$ associated with x_n. Thus, the values assigned to $\{w_i | i \in \Delta(\Sigma_k)\}$ in the row-generating PSP are calculated as

$$\begin{pmatrix} \mathbf{c}_x - \mathbf{c}_{\mathcal{B}}\mathbf{B}^{-1}\mathbf{F} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{D_2} & \mathbf{0} \\ \mathbf{D_3} & -\mathbf{I} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{c}_x - \mathbf{c}_{\mathcal{B}}\mathbf{B}^{-1}\mathbf{F} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{D_2}^{-1} & \mathbf{0} \\ \mathbf{D_3}\mathbf{D_2}^{-1} & -\mathbf{I} \end{pmatrix} \tag{4.15}$$

$$= \begin{pmatrix} (\mathbf{c}_x - \mathbf{c}_{\mathcal{B}}\mathbf{B}^{-1}\mathbf{F})\mathbf{D_2}^{-1} & \mathbf{0} \end{pmatrix}, \tag{4.16}$$

which are identical to those computed in (4.14) based on $\mathbf{B}_k$.

From the optimality of $\mathrm{SRMP}(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$, $c_l - \sum_{j \in J} A_{jl} u_j - \sum_{i \in I(\bar{K}, \bar{N})} C_{il} w_i \geq 0$ for a variable $y_l, l \in \bar{K}$ and $d_n - \sum_{m \in M} B_{mn} v_m - \sum_{i \in I(\bar{K}, \bar{N})} D_{in} w_i \geq 0$ for a variable $x_n, n \in \bar{N}$. Furthermore, no variable in $\mathrm{SRMP}(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ is present in a linking constraint $i \in \Delta_x(\Sigma_k)$, and $w_i = 0$ for all $i \in \Delta_s(\Sigma_k)$. Thus, we conclude that $\bar{c}_l = c_l - \sum_{j \in J} A_{jl} u_j - \sum_{i \in I(\bar{K}, \bar{N})} C_{il} w_i - \sum_{i \in \Delta(\Sigma_k)} C_{il} w_i \geq 0$ for a variable $y_l, l \in \bar{K}$ and $\bar{d}_n = d_n - \sum_{m \in M} B_{mn} v_m - \sum_{i \in I(\bar{K}, \bar{N})} D_{in} w_i - \sum_{i \in \Delta(\Sigma_k)} D_{in} w_i \geq 0$ for a variable $x_n, n \in \bar{N}$ in $\mathrm{SRMP}(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$. For $x_n, n \in S_N(\Sigma_k)$, $\bar{d}_n = d_n - \sum_{m \in M} B_{mn} v_m - \sum_{i \in I(\bar{K}, \bar{N})} D_{in} w_i - \sum_{i \in \Delta(\Sigma_k)} D_{in} w_i = d_n - \sum_{m \in M} B_{mn} v_m - \sum_{i \in \Delta(S_K^k)} D_{in} w_i \geq 0$ by (4.11b) and because the values of the dual variables $\{w_i | i \in \Delta(\Sigma_k)\}$ employed in the row-generating PSP are optimal with respect to $\mathrm{SRMP}(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$. Thus, we arrive at the conclusion that the values of the dual variables calculated in (4.14) are dual feasible as required. □

The proof of Lemma 4.1.4 establishes formal arguments for some fundamental claims and propositions that we employed in the development of our column-and-row generation approach for CDR-problems. These are summarized in the following corollary.

Corollary 4.1.1

a. *The optimal values of the dual variables* $\{u_j | j \in J\}$, $\{v_m | m \in M\}$, *and* $\{w_i | i \in I(\bar{K}, \bar{N})\}$ *are identical for* $\mathrm{SRMP}(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ *and* $\mathrm{SRMP}(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$.

b. The values assigned to the dual variables $\{w_i|i \in \Delta(\Sigma_k)\}$ *in the row-generating PSP are optimal for* $\mathrm{SRMP}(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$. *For* $i \in \Delta_s(\Sigma_k)$, *we have* $w_i = 0$ *at optimality.*

c. The reduced costs of $\{y_l|l \in \bar{K}\}$ *and* $\{x_n|n \in \bar{N}\}$ *are identical with respect to the optimal dual solutions of* $\mathrm{SRMP}(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ *and* $\mathrm{SRMP}(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$.

d. The reduced cost $\bar{c}_k$ *computed in the row-generating PSP for any* $y_k, k \notin \bar{K}$ *and* $\mathcal{F}_k \in \mathcal{P}_k$ *is equal to the reduced cost of* y_k *with respect to the optimal solution of* $\mathrm{SRMP}(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$.

e. A variable $x_n, n \in S_N(\Sigma_k)$ *basic at the optimal solution of* $\mathrm{SRMP}(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$ *is equal to zero.*

We are now ready to prove that the column-and-row generation algorithm depicted in Figure 4.1 is an optimal algorithm for solving (MP) for CDR-problems characterized by Assumptions 3.1.1-3.1.3.

THEOREM 4.1.1 *Given an optimal basis* $\mathbf{B}$ *for* $\mathrm{SRMP}(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ *and a set of associated optimal values for the dual variables* $\{u_j|j \in J\}$, $\{v_m|m \in M\}$, *and* $\{w_i|i \in I(\bar{K}, \bar{N})\}$, *the proposed column-and-row generation algorithm terminates with an optimal solution for the master problem (MP) if* $\zeta_y \geq 0$, $\zeta_x \geq 0$, *and* $\zeta_{yx} \geq 0$ *in three consecutive calls to the* $y-$, $x-$, *and the row-generating PSPs, respectively.*

PROOF. According to the flow of the proposed column-and-row generation algorithm in Figure 4.1, we invoke the row-generating PSP only after the $y-$ and $x-$PSPs fail to identify negatively priced $y-$ and $x-$ variables, respectively, given the optimal dual solution of $\mathrm{SRMP}(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$. If in addition the optimal objective value ζ_{yx} of the row-generating PSP is nonnegative given the optimal dual solution of $\mathrm{SRMP}(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$, then the algorithm terminates with an optimal solution to (MP) as we prove below. On the other hand, if the row-generating PSP terminates at least once successfully with a negatively priced new $y-$variable, then the optimal values of the dual variables are updated by re-optimizing SRMP augmented with new rows and columns. Therefore, in this case we cannot claim optimality following a subsequent unsuccessful optimization of the row-generating PSP with $\zeta_{yx} \geq 0$, and we must call the $y-$ and $x-$PSPs again (FLAG=1 at the termination of the row-generating PSP in Figure 4.1).

Now, assume that $\zeta_y \geq 0$, $\zeta_x \geq 0$, and $\zeta_{yx} \geq 0$ in three consecutive calls to the $y-$, $x-$, and the row-generating PSPs, respectively. In the sequel, we show that $\bar{c}_k \geq 0$ for $k \in K$ and $\bar{d}_n \geq 0$ for $n \in N$ even if we introduce the currently absent set of linking constraints $I \setminus I(\bar{K}, \bar{N})$ into the SRMP. Recall that the linking constraints feature a block-diagonal form, where a block is defined by a set of $x-$ and surplus variables that only appear in this block. For each block, we can choose any extreme point of a polyhedron similar to that defined by (4.11b)-(4.11d), designate new $x-$ and surplus variables as basic as prescribed, and then incorporate these linking constraints and the associated $x-$ and surplus variables in the SRMP. The resulting SRMP and the associated basis are denoted as SRMP$(\bar{K}, N, I)$ and $\mathbf{B}'$, respectively. Moreover, recall that Lemma 4.1.3 does not require that (4.11) is solved to optimality. Thus, we can develop a proof analogous to that of Lemma 4.1.4 and show that $\mathbf{B}'$ is an optimal basis for SRMP$(\bar{K}, N, I)$. Clearly, $\bar{d}_n \geq 0, n \in N$ and $\bar{c}_l \geq 0, l \in \bar{K}$ hold at the optimal solution of SRMP$(\bar{K}, N, I)$, and we have $x_n = 0$ for all $N \setminus \bar{N}$ by a straightforward extension of Corollary 4.1.1e. Finally, in order to complete the proof we need to argue that no variable $y_k, k \in K \setminus \bar{K}$ prices out favorably with respect to the optimal solution of SRMP$(\bar{K}, N, I)$. To this end, note that all missing linking constraints $\Delta(S_K^k)$ induced by all minimal variable sets of the form $\{y_l | l \in S_K^k\}$ are included in SRMP$(\bar{K}, N, I)$. The corresponding family of index sets is clearly an element of $\mathcal{P}_k$. Thus, we conclude that the reduced cost $\bar{c}_k$ of y_k is nonnegative because $0 \leq \zeta_{yx} \leq \bar{c}_k$, where ζ_{yx} denotes the minimum reduced cost of y_k computed over all possible members of $\mathcal{P}_k$ in the row-generating PSP in (4.10). □

4.2 Applications of The Proposed Method

In this section, the proposed solution method is applied to our illustrative problems, MSCS, QSC, and TCR. For the TCR problem, we demonstrate that the optimality condition proposed in [2] is incorrect and may lead to a suboptimal solution at termination. We identify the source of this error and discuss how our proposed column-and-row generation algorithm may be applied to this TCR problem in order to solve the proposed large-scale LP correctly. Moreover, we will conduct computational experiments on the example problems except for the TCR problem. The computational experiments for the TCR problem are presented in the next chapter where the extension with Lagrangian relaxation is given so that a comparison of the algorithms can be made. We conducted our computational experiments on a single core of an HP Compaq DX 7400 computer with a 2.40 GHz Intel Core 2 Quad Q6600 CPU and 3.25 GB of RAM running on Windows XP. Our codes are implemented in Visual C++, and IBM ILOG CPLEX 12.1/ Concert Technology 2.9 is employed to solve the linear programs in the column generation procedure. All of the instances generated for the example problems are also solved optimally by CPLEX 12.1.

4.2.1 Multi-Stage Cutting Stock Problem

We develop the subproblems for a column-and-row generation algorithm that solves the LP relaxation of the one-dimensional MSCS problem given in (3.2)-(3.5) with exponentially many cutting patterns both in the first and second stages. We follow the steps of the generic framework for CDR-problems developed in Section 4.1.

As discussed in Section 3.2, MSCS is a CDR-problem with no interaction, that is, the cardinality of a minimal variable set is just one. A first-stage cutting pattern represented by y_k is generated by any feasible combination of existing and new intermediate rolls. For each new intermediate roll currently absent from the SRMP included in the pattern, a single new linking constraint is introduced into the SRMP. Thus, $\mathcal{F}_k = \{\{k\}\}$, $\Sigma_k = \{k\}$, and $\Delta(\Sigma_k)$ denotes the set of linking constraints that correspond to the new intermediate rolls in the pattern. All three types of PSPs introduced in Section 4.1 are required for the MSCS problem. In the sequel, we explain how each PSP is constructed. To this end, we first state the dual of the LP (3.2)-(3.5):

$$\begin{aligned}
\text{maximize} \quad & \sum_{m\in M} b_m v_m, && (4.17)\\
\text{subject to} \quad & \sum_{i\in I} C_{ik} w_i \leq 1, \quad k \in K, && (4.18)\\
& \sum_{m\in M} B_{mn} v_m + \sum_{i\in I} D_{in} w_i \leq 0, \quad n \in N, && (4.19)\\
& v_m \geq 0, m \in M, \quad w_i \geq 0, i \in I, && (4.20)
\end{aligned}$$

where $\{v_m | m \in M\}$, and $\{w_i | i \in I\}$ are the dual variables corresponding to the primal constraints (3.3) and (3.4), respectively. Recall that a single non-zero entry $D_{in} = -1$ in column n of D indicates that the cutting pattern n for the second stage is cut from the intermediate roll i. This implies for x_n associated with a cutting pattern obtained from the intermediate roll i that the inequality (4.19) reduces to $\sum_{m\in M} B_{mn} v_m \leq w_i$.

In the $y-$PSP given below, the objective is to identify a violated constraint (4.18) for a first-stage cutting pattern composed of the set of intermediate rolls $I(\bar{K}, \bar{N})$ present in the current SRMP:

$$\begin{aligned}
\text{maximize} \quad & \sum_{i\in I(\bar{K},\bar{N})} w_i C_i, \\
\text{subject to} \quad & \sum_{i\in I(\bar{K},\bar{N})} \epsilon_i C_i \leq W, \qquad (4.21)\\
& C_i \in \mathbb{Z}^+ \cup \{0\}, i \in I(\bar{K}, \bar{N}),
\end{aligned}$$

where W is the stock roll width, ϵ_i is the width of the intermediate roll i, and C_i is the number of times intermediate roll i is cut from the stock roll. Clearly, the $y-$PSP is an integer knapsack problem that may be solved efficiently by well-known methods in the literature. If the optimal objective function value of (4.21) is larger than 1, then a new first-stage cutting pattern with a negative reduced cost is added to the SRMP. Representing this pattern by y_k, we have $\bar{K} \leftarrow \bar{K} \cup \{k\}$.

In the $x-$PSP, we search for a second-stage cutting pattern with a negative reduced cost that is cut from one of the existing intermediate rolls in the current SRMP.

In other words, we determine whether one of the dual constraints $\sum_{m \in M} B_{mn} v_m \leq w_i, i \in I(\bar{K}, \bar{N})$, is violated:

$$\begin{aligned} \text{maximize} \quad & \sum_{m \in M} v_m B_m, \\ \text{subject to} \quad & \sum_{m \in M} \pi_m B_m \leq \epsilon_i - e^{min}, \\ & B_m \in \mathbb{Z}^+ \cup \{0\}, m \in M, \end{aligned} \tag{4.22}$$

where i is the index for the existing intermediate roll of width ϵ_i under consideration, π_m is the width of the finished roll $m \in M$, and B_m denotes the number of times finished roll m is cut from intermediate roll i. The parameter e^{min} represents a mandatory minimal edge for intermediate rolls (see [72, 73]). Similar to the previous case, the $x-$PSP is an integer knapsack problem. If the optimal objective function value of (4.22) is larger than w_i, then a new second-stage cutting pattern with a negative reduced cost is added to the SRMP. Representing this pattern by x_n, we have $\bar{N} \leftarrow \bar{N} \cup \{n\}$.

In the row-generating PSP, a first-stage cutting pattern may contain new intermediate rolls in addition to the existing ones. Following the structure of the general formulation in (4.10)-(4.11), the row-generating PSP for MSCS is stated as:

$$\text{maximize} \quad \sum_{i \in I(\bar{K}, \bar{N})} w_i C_i + \sum_{i' \in I \setminus I(\bar{K}, \bar{N})} \alpha_{i'}, \tag{4.23a}$$

$$\text{subject to} \quad \sum_{i \in I(\bar{K}, \bar{N})} \epsilon_i C_i + \sum_{i' \in I \setminus I(\bar{K}, \bar{N})} \epsilon_{i'} C_{i'} \leq W, \tag{4.23b}$$

$$C_i \in \mathbb{Z}^+ \cup \{0\}, \qquad i \in I, \tag{4.23c}$$

where

$$
\begin{aligned}
\alpha_{i'} = &\text{maximize} && w_{i'} C_{i'}, && && \text{(4.24a)} \\
&\text{subject to} && \sum_{m \in M} \pi_m B_m \leq \epsilon_{i'} - e^{min}, && && \text{(4.24b)} \\
& && \epsilon^{min} \leq \epsilon_{i'} \leq \epsilon^{max}, && && \text{(4.24c)} \\
& && B_m \in \mathbb{Z}^+ \cup \{0\}, && m \in M, && \text{(4.24d)} \\
& && \sum_{m \in M} v_m B_m \leq w_{i'}, && && \text{(4.24e)} \\
& && w_{i'} \geq 0, && && \text{(4.24f)} \\
& && \text{At least one of (4.24e) or (4.24f) is tight.} && && \text{(4.24g)}
\end{aligned}
$$

Due to their potential size, we cannot explicitly generate the sets of all feasible first- and second-stage cutting patterns. Therefore, the structural constraints (4.23b)-(4.23c) and (4.24b)-(4.24d) that define the feasible first- and second-stage cutting patterns, respectively, are incorporated in the row-generating PSP. The constraints (4.23b) and (4.24b) are the classical knapsack constraints for the first- and second-stage cutting patterns, respectively. The constraint (4.24c) defined in [73] imposes lower and upper bounds on the width of a new intermediate roll. In [73], it is also imposed that (4.24b) is satisfied with equality, that is, the width of any unknown intermediate roll may be presented as a linear combination of finished roll widths plus minimum edge. The constraint (4.24e) corresponds to (4.11b) and mandates that the unknown dual variable associated with the currently absent intermediate roll i' is larger than or equal to the sum of the dual variables for the finished rolls that are cut from this intermediate roll. Note that all new linking constraints in this problem are of type (3.1) as specified in Assumption 3.1.3. Therefore, $\Delta_+(\{k\}) = \Delta(\{k\})$ and $\Delta_0(\{k\}) = \emptyset$ which only requires $w_{i'} \geq 0$ as stated in (4.24f). Finally, constraint (4.24g) is the counterpart of (4.11e). Observe that in the above row-generating PSP, we assume no limit on the number of new intermediate rolls in contrast to [73, 72] which heuristically limit the number of new intermediate rolls to one when solving this subproblem.

Note that for each $i' \in I \setminus I(\bar{K}, \bar{N})$, the constraints in (4.24) are distinct. Also, (4.23b) is the constraint that links $I \setminus I(\bar{K}, \bar{N})$ to $I(\bar{K}, \bar{N})$. This subproblem can be solved by column generation through forming the integer master problem with the constraints (4.23a)-(4.23c). The rest of the constraints (4.24b)-(4.24g) form the subproblem that characterizes the unknown intermediate rolls, the $x-$variables that are cut from these rolls, and also the values of the unknown dual variables. This problem structure is known as the nested decomposition or two-level decomposition, since the row-generating PSP itself is also solved by column generation (see [31] and [70] for the details of the two-level and nested decomposition procedures). We first relax the integrality constraints (4.23c) in the integer master problem and then replace I by $I(\bar{K}, \bar{N})$ to form the restricted master problem of the row-generating PSP, which we refer to as RMPS, given by

$$z_{RMPS} = \text{maximize} \quad \sum_{i \in I(\bar{K}, \bar{N})} w_i C_i, \tag{4.25}$$

$$\text{subject to} \quad \sum_{i \in I(\bar{K}, \bar{N})} \epsilon_i C_i \leq W, \tag{4.26}$$

$$C_i \geq 0, \qquad i \in I(\bar{K}, \bar{N}). \tag{4.27}$$

Suppose that ξ is the dual variable corresponding to the constraint (4.26). The pricing subproblem of the row-generating PSP, which we refer to as SPSP, is

$$z_{SPSP} = \text{maximize} \quad w - \xi\epsilon, \tag{4.28}$$

$$\text{subject to} \quad \sum_{m \in M} \pi_m B_m = \epsilon - e^{min}, \tag{4.29}$$

$$\epsilon^{min} \leq \epsilon \leq \epsilon^{max}, \tag{4.30}$$

$$B_m \in \mathbb{Z}^+ \cup \{0\}, \qquad m \in M, \tag{4.31}$$

$$\sum_{m \in M} v_m B_m \leq w, \tag{4.32}$$

$$w \geq 0, \tag{4.33}$$

$$\epsilon \geq 0, \tag{4.34}$$

$$\text{At least one of (4.32) or (4.33) is tight.} \tag{4.35}$$

Observe that (4.24b) turns into (4.29) as it is proved in [73]. The value of the dual variable (w), the width of the new intermediate roll (ϵ), and the second-stage cutting pattern that is cut from this intermediate roll are determined by this subproblem. Constraint (4.35) imposes that either $w = 0$ or $w = \sum_{m \in M} v_m B_m$. Since $v_m \geq 0$ and $B_m \geq 0$ for all $m \in M$, if (4.33) is tight, we must have $\sum_{m \in M} v_m B_m = w = 0$. Thus, (4.32) can be replaced by $\sum_{m \in M} v_m B_m = w$ without loss of generality. This is equivalent to adding the corresponding $x-$variable to the basis. Plugging the terms w and ϵ into subproblem (4.28)-(4.35), we obtain:

$$
\begin{aligned}
z_{SPSP} = \text{maximize} \quad & \sum_{m \in M} (v_m - \xi \pi_m) B_m - \xi e^{min}, \\
\text{subject to} \quad & \epsilon^{min} \leq \sum_{m \in M} \pi_m B_m + e^{min} \leq \epsilon^{max}, \\
& B_m \in \mathbb{Z}^+ \cup \{0\}, \qquad\qquad m \in M.
\end{aligned}
\tag{4.36}
$$

If $v_m \leq \xi \pi_m$ for all $m \in M$, the optimal objective function value cannot be positive and the column generation at this level terminates. Problem (4.36) generates a new intermediate roll and a second-stage cutting pattern that can be cut from this intermediate roll. If $z_{SPSP} > 0$, this new intermediate roll is added to $I(\bar{K}, \bar{N})$ and (4.25)-(4.27) is solved again.

To solve the row-generating PSP, a branch-and-price algorithm must be developed (see [8] for a comprehensive survey on the branch-and-price algorithm). If the result of the branch-and-price algorithm is larger than one, a first-stage cutting pattern is added to the SRMP along with a set of linking constraints and $x-$variables. When SPSP cannot generate a positive reduced cost column, we must branch to eliminate the current fractional solution and reoptimize the LP relaxation at each node in the branch-and-bound tree. In our case, the master problem is a knapsack problem and the pricing subproblem is a knapsack problem with a lower-bounding constraint. Moreover, the solution of the master problem RMPS is trivial since the continuous relaxation of the knapsack problem is readily available after putting the variables in a nonincreasing order of $(w_i/\epsilon_i), i \in I(\bar{K}, \bar{N})$. Without loss of generality, let the indices $i_1, i_2, \ldots$ of the variables correspond to their position in this ordering. At the root node, $C_{i_1} = W/\epsilon_{i_1}$,

$C_i = 0$ for $i \in I(\bar{K}, \bar{N}) \backslash i_1$ and $z_{RMPS} = w_{i_1} W/\epsilon_{i_1}$. The value of the dual variable corresponding to the knapsack constraint is $\xi = (w_{i_1}/\epsilon_{i_1})$. The pricing subproblem checks whether $z_{SPSP} > 0$, which implies

$$\begin{aligned} \sum_{m \in M} v_m B_m - \xi \left(\sum_{m \in M} \pi_m B_m + e^{min} \right) &> 0, && (4.37) \\ w - \xi \epsilon &> 0, && (4.38) \\ \frac{w}{\epsilon} &> \xi, && (4.39) \\ \frac{w}{\epsilon} &> \frac{w_{i_1}}{\epsilon_{i_1}}. && (4.40) \end{aligned}$$

If so, there exists $i \in I \setminus I(\bar{K}, \bar{N})$ for which $(w_i/\epsilon_i) > (w_{i_1}/\epsilon_{i_1})$. If there is no such intermediate roll and the solution of the RMPS is fractional, then branching occurs.

Suppose that a variable i^* for which $(w_{i^*}/\epsilon_{i^*}) > (w_{i_1}/\epsilon_{i_1})$ is detected. The intermediate roll i^* is added to the RMPS, and since it is now the variable with the largest ratio (w/ϵ), we set $C_{i^*} = W/\epsilon_{i^*}$, $C_i = 0$ for $i \in I(\bar{K}, \bar{N}) \backslash i^*$ and $z_{RMPS} = w_{i^*} W/\epsilon_{i^*}$. Since i^* is the variable with the largest ratio already generated by SPSP, solving SPSP again does not generate a new intermediate roll having a ratio larger than w_{i^*}/ϵ_{i^*}. Hence, the column generation terminates. If W/ϵ_{i^*} is fractional, the most general branching rule can be applied, where we create two nodes in which we impose $C_{i*} \leq \lfloor W/\epsilon_{i*} \rfloor$ and $C_{i*} \geq \lceil W/\epsilon_{i*} \rceil$. In the former branch, let ξ_{i*} be the dual variable corresponding to $C_{i*} \leq \lfloor W/\epsilon_{i*} \rfloor$. When RMPS is solved in the former branch, we have $C_{i^*} = \lfloor W/\epsilon_{i*} \rfloor$, $C_{i_1} = (W - \epsilon_{i*} \lfloor W/\epsilon_{i*} \rfloor)/\epsilon_{i_1}$, $C_i = 0$ for $i \in I(\bar{K}, \bar{N}) \backslash \{i^*, i_1\}$, and $\xi = (w_{i_1}/\epsilon_{i_1})$ and $\xi_{i*} = (w_{i*} - \epsilon_{i*} w_{i_1}/\epsilon_{i_1})$. Then, we check whether there is any intermediate roll for which $(w_i/\epsilon_i) > (w_{i_1}/\epsilon_{i_1})$. The intermediate roll i^*, whose reduced cost is nonnegative in RMPS, is generated again by the SPSP since SPSP does not take into consideration the dual variable corresponding to the bounding constraint, ϵ_{i*}. Therefore, we have to add constraints to the SPSP to prevent the regeneration of intermediate roll i^*. Similar difficulties pertaining to such branching are defined in [67]. To overcome this difficulty, we convert SPSP into a 0-1 problem using the transformation given in [59] that is designed for knapsack problems. To this end, we put the natural upper bounds, ub_m, $m \in M$ on the items such that $B_m \leq \lfloor (\epsilon^{max} - e^{min})/\pi_m \rfloor = ub_m$. For each variable

$B_m, m \in M$, a sequence of $\lceil log_2(ub_m + 1)\rceil$ binary variables, say B_{m_j}, is introduced. Suppose that $\hat{B}_{m_j}$ for $m \in M$ and $j = 1, \ldots, \lceil log_2(ub_m + 1)\rceil$ are the values of the binary variables resulting from the solution of the transformed SPSP. To rule out the generation of variable i^*, we add boolean constraints of the following form

$$\sum_{m=1}^{M}\left(\sum_{j:\hat{B}_{m_j}=1}(1 - B_{m_j}) + \sum_{j:\hat{B}_{m_j}=0} B_{m_j}\right) \geq 1. \tag{4.41}$$

There may be several alternate solutions of the 0-1 SPSP that result in the intermediate roll i^*. In this case, a set of constraints of the form (4.41) is added to the 0-1 SPSP to prevent the generation of i^*. As we go deeper in the branch-and-bound tree, the number of added constraints increases.

In summary, at some node, where $\xi = w_{i_n}/\epsilon_{i_n}$ and n is the position of the given intermediate roll in the nonincreasing sequence of w/ϵ values, the column generation algorithm generates intermediate roll i for which $w_i/\epsilon_i > w_{i_n}/\epsilon_{i_n}$. When no such intermediate roll is generated by SPSP, branching occurs if the solution is fractional. At any node in the tree, there is only one fractional variable due to the structure of the RSMP. At a particular node, assume that C_{i_n} is the fractional variable and let its value be $\bar{C}_{i_n}$. On the branch of the form $C_{i_n} \leq \lfloor \bar{C}_{i_n} \rfloor$, the value of ξ is set to $w_{i_{n+1}}/\epsilon_{i_{n+1}}$, which is the largest ratio smaller than w_{i_n}/ϵ_{i_n}, and the SPSP is called. However, on the branch of the form $C_{i_n} \geq \lceil \bar{C}_{i_n} \rceil$, the value of ξ is set to $w_{i_{n-1}}/\epsilon_{i_{n-1}}$ unless the node problem becomes infeasible. If $\bar{C}_{i_n}$ is not fractional, we update the best upper-bound (UB) when $z_{RMPS} < UB$. Pruning rule is applied when $z_{RMPS} > UB$.

We apply the branch-and-price algorithm explained above to the instances given in [73, 72]. For the problem instance in [73], we observe that the proposed branch-and-price algorithm finds the optimal objective function value of the LP relaxation of the MSCS problem which is 20. The same value is also reached by the algorithm given in [73]. However, for the problem instance given in [72], our algorithm reaches the optimal objective function value of the LP relaxation of the MSCS problem which is 35.97, while the algorithm given in [72] stops at 36.

Instead of applying a full-blown branch-and-price algorithm for solving the row-generating PSP, we construct a heuristic algorithm mimicking the routine of the branch-and-price algorithm explained above. This algorithm is detailed in Algorithm 1. At the beginning of the algorithm, we have a set of intermediate rolls denoted by $I(\bar{K}, \bar{N})$, and the ratios (w/ϵ) are stored in a nonincreasing order in array R (line 1, Algorithm 1). We first set ξ to the ratio of the corresponding intermediate roll (line 3, Algorithm 1). Then, we solve SPSP, which is a 0-1 knapsack problem with additional constraints, to generate an intermediate roll $(\hat{i})$ whose ratio $(w_{\hat{i}}/\epsilon_{\hat{i}})$ is larger than ξ (line 5, Algorithm 1). If $z_{SPSP} > 0$, we add a constraint to RMPS in order to prevent the generation of this intermediate roll again (line 7, Algorithm 1) and add the generated intermediate roll to $I(\bar{K}, \bar{N})$ (line 8, Algorithm 1). If the number of intermediate rolls added at iteration i of the for-loop is a multiple of a parameter S, we solve RMPS as an integer program (line 11, Algorithm 1). If $z_{SPSP} \leq 0$, we check whether any intermediate rolls are added after the last call of RMPS, in which case RMPS is solved as an integer program (line 20, Algorithm 1). If the optimal objective function value of integer RMPS is larger than one, then we stop the algorithm and add the resulting cutting pattern for the first stage to SRMP. Otherwise, the counter is set to zero and the same operations are repeated for $i + 1$. When the main for-loop ends after $i = |R|$, the column-and-row generation algorithm terminates, since no cutting pattern for the first stage with reduced cost larger than one is detected by our heuristic algorithm.

Solving SPSP becomes harder when the number of intermediate rolls increases, since we add a new constraint for each intermediate roll. Since SPSP may generate many intermediate rolls for some iteration i, we choose to solve RMPS as an integer program at given intervals to check whether a cutting pattern for the first stage with positive reduced cost exists in the current solution.

Moreover, the optimal integer solution of RMPS may actually use only a small number of new intermediate rolls that are generated by SPSP. We may delete the new intermediate rolls, which do not exist in the first-stage cutting pattern. However, this also requires the deletion of the associated constraints in SPSP that prevent the generation of these rolls. However, such a strategy takes more time in our computational experiments because in the next call of the row-generating PSP, these intermediate rolls

Algorithm 1: `Algorithm to solve the row-generating PSP for MSCS problem.`

Input: $I(\bar{K}, \bar{N}), R, counter = 0$
for $i = 1$ **to** $|R|$ **do**
    $\xi = R(i)$;
    while *true* **do**
        Solve $SPSP \rightarrow (z_{SPSP}, \hat{i})$;
        if $z_{SPSP} > 0$ **then**
            add (4.41) for $\hat{i}$;
            $I(\bar{K}, \bar{N}) \leftarrow I(\bar{K}, \bar{N}) \cup \hat{i}$;
            $counter \rightarrow counter + 1$;
            if $counter = S$ **then**
                Solve $RMPS^{IP}$;
                if $z_{RMPS} > 1$ **then**
                    break out of the for loop;
                $counter = 0$;
        else
            break out of the while loop;
    if $counter > 0$ **then**
        Solve $RMPS^{IP}$;
        if $z_{RMPS} > 1$ **then**
            break out of the for loop;
    $counter = 0$;

may be generated again with their corresponding prevention constraints of the form (4.41).

In this section, we solve the instances given in [73, 72], and eight other randomly generated instances by Algorithm 1 where $S = 4$ is selected. To generate the set of instances, we determine pseudo-random integral numbers in a given range, which are generated by the operator $R(m, n)$ where m and n are the limits of the range, as follows: $W = 100R(10, 110)$, $|M| = R(3, 7)$, $b_m = R(20, 200)$ for $m \in M$, $e^{min} = R(10, 50)$, $\epsilon^{min} = R(\frac{W}{11}, \frac{W}{9})$, $\epsilon^{max} = R(\frac{W}{5}, \frac{W}{3})$, and $\pi_m = R(\frac{W}{20}, \frac{W}{12})$ for $m \in M$. To compare our results, we implement the algorithm given in [73, 72]. Moreover, we develop an optimal approach to solve the LP relaxation of the MSCS problem in which all possible intermediate rolls are enumerated a priori and the cutting patterns for the first and the second stages are generated by $y-$ and $x-$PSPs.

In [73, 72], the illustrative problems are solved by starting with the intermediate roll ϵ^{min} which may not exist in the list of intermediate rolls that can be generated by SPSP ($\epsilon^{min} \neq \sum_{m \in M} \pi_m B_{m.} + e^{min}$). Also, in [73, 72], an algorithm is given to generate a *good initial solution* including the intermediate roll ϵ^{min}. The initial solution we choose to start with corresponds to this *good initial solution* except for the intermediate roll ϵ^{min}. The *good initial solution* is found as follows: For each finished roll width π_m, we identify an intermediate roll width ϵ as

$$a = \lfloor (\epsilon^{max} - e^{min}) / \pi_m \rfloor \tag{4.42}$$

$$s = e^{min} + a\pi_m \tag{4.43}$$

Clearly, $\epsilon \leq \epsilon^{max}$ is guaranteed. If $\epsilon \geq \epsilon^{min}$, this intermediate roll is added to the set $I(\bar{K}, \bar{N})$. If $I(\bar{K}, \bar{N}) = \emptyset$ after each finished roll is processed, $I(\bar{K}, \bar{N}) = \{\epsilon^{min}\}$.

In Table 4.2, the results of the experiments are given. The instances are defined based on their number of finished rolls (# Fin. Rolls) and their stock roll width (W). The first two instances are taken from [73] and [72], respectively. The columns show the results of the optimal solution (Optimal), our heuristic column-and-row generation algorithm (HCRG), and the algorithm presented in [73] (Zak). We report the objective function value (OFV), the total number of intermediate rolls generated (# Int. Rolls), and the solution time (Time). Except for the second instance, HCRG reaches the optimal objective function values of the instances given in Table 4.2. On the other hand, the algorithm given in [73] reaches the objective function values in only four instances. HCRG, on average, generates 80% of all possible intermediate rolls. On the other hand, the number of intermediate rolls that are generated by Zak is approximately eight times smaller than that generated by HCRG and one tenth of all possible intermediate rolls.

The solution times of the algorithms seem to be proportional to the number of intermediate rolls generated but not the number of finished rolls. For example, although the number of finished rolls is 6 in the sixth instance, its solution time is considerably larger than that of the ninth instance where there are only 7 finished rolls. The solution time of HCRG is on average 17% of that of the optimal approach. However, the solution time of Zak is even lower than that of HCRG. With only a small deviation from the optimal solution, Zak proves to be a very efficient heuristic approach. As explained previously, HCRG mimics the branch-and-price algorithm so that it is more geared towards reaching the optimal solution rather than finding a solution in a short amount of time.

Table 4.2: Comparison of algorithms on MSCS test instances.

Instances		**Optimal**			**HCRG**			**Zak**		
# Fin. Rolls	W	OFV	# Int. Rolls	Time	OFV	# Int. Rolls	Time	OFV	# Int. Rolls	Time
4	5,000	35.97	47	32.76	35.97	37	23.27	35.97	11	7.46
2	194	20.00	3	1.23	20.89	1	0.09	20.89	1	0.20
3	2,000	17.78	18	16.25	17.78	14	12.72	18.00	3	2.50
4	3,100	545.68	64	63.50	545.68	48	45.78	545.70	13	12.47
4	6,000	46.60	64	92.36	46.60	25	30.51	46.60	11	8.64
6	10,400	556.13	481	30,015.80	556.13	395	3,715.06	556.39	21	55.00
4	3,300	510.94	65	23.11	510.94	51	69.18	510.94	8	5.61
4	5,800	361.99	67	168.08	361.99	48	74.17	361.99	14	11.72
7	9,400	860.03	268	3,059.55	860.03	240	1,983.05	860.27	19	41.91
5	5,100	54.60	38	24.30	54.60	20	19.81	54.83	7	5.33
Average		300.97	111.50	3,349.69	301.06	87.90	597.36	301.16	10.80	15.08

4.2.2 Quadratic Set Covering

In this section, we develop the subproblems for a column-and-row generation algorithm that solves the LP relaxation of the QSC problem which belongs to the class of CDR-problems with interaction. The cardinality of each minimal variable set for this problem is two, and three linking constraints of type (3.8)-(3.10) and one auxiliary $x-$variable are associated with a minimal variable set. As described in Section 3.2, the set of all $y-$variables is given explicitly, and the set of all possible pairs composed of the $y-$variables is denoted by P.

To solve the formulation (3.6)-(3.12) by column-and-row generation, we initialize the SRMP with a set of columns $\bar{K}$ that satisfy the covering constraints (3.7) in addition to the set of all linking constraints (3.8)-(3.10) induced by $\{y_l | l \in \bar{K}\}$. A new variable y_k is always added to the SRMP along with three linking constraints of type (3.8)-(3.10) and a variable x_{kl} for each pair of variables y_k, y_l, where $l \in \bar{K}$. Thus, the column-and-row generation mechanism maintains that the SRMP is constituted by $\{y_l | l \in \bar{K}\}$, $\{x_n | n \in S_N(\bar{K})\}$, and the set of all linking constraints $\Delta(\bar{K})$ induced by $\bar{K}$ at all times during the course of the algorithm. Moreover, the $x-$variables are auxiliary and only appear in the linking constraints. Therefore, for QSC we only need to solve the row-generating PSP developed below. We first state the dual of the LP (3.6)-(3.12):

$$\begin{aligned}
\text{maximize} \quad & \sum_{j \in J} u_j + \sum_{(k,l) \in P, k<l} w_{kl}, \\
\text{subject to} \quad & \sum_{j \in J} A_{jk} u_j + \sum_{(k,l) \in P, k<l} (w_{kl} + \varphi_{kl}) + \sum_{(l,k) \in P, l<k} (w_{lk} + \varphi'_{lk}) \leq f_{kk}, \;\; k \in K, \\
& - w_{kl} - \varphi_{kl} - \varphi'_{kl} \leq 2 f_{kl}, \quad (k,l) \in P, k<l, \\
& \varphi_{kl}, \varphi'_{kl} \geq 0, w_{kl} \leq 0, \quad (k,l) \in P, k<l,
\end{aligned}$$

where $\{u_j | j \in J\}$ are the dual variables corresponding to the covering constraints (3.7), and w_{kl}, φ_{kl}, φ'_{kl} for $(k,l) \in P, k<l$, are the dual variables associated with the linking constraints (3.8)-(3.10), respectively.

Following the structure of the general formulation in (4.10)-(4.11), the row generating PSP for QSC is stated as:

$$\zeta_{yx} = \min_{k \in (K \setminus \bar{K})} \left\{ f_{kk} - \sum_{j \in J} A_{jk} u_j - \max_{\mathcal{F}_k \in \mathcal{P}_k} \left(\sum_{\{k,l\} \in \mathcal{F}_k, k<l} \alpha_{kl} + \sum_{\{l,k\} \in \mathcal{F}_k, l<k} \alpha_{lk} \right) \right\}, \tag{4.44}$$

where

$$\begin{aligned}
\alpha_{kl} = \text{maximize} \quad & w_{kl} + \varphi_{kl}, && (4.45a)\\
\text{subject to} \quad & -w_{kl} - \varphi_{kl} - \varphi'_{kl} \leq 2f_{kl}, && (4.45b)\\
& w_{kl} = \varphi'_{kl} = 0, && (4.45c)\\
& \varphi_{kl} \geq 0, && (4.45d)\\
& \text{At least one of (4.45b) or (4.45d) is tight,} && (4.45e)\\
\alpha_{lk} = \text{maximize} \quad & w_{lk} + \varphi'_{lk}, && (4.45f)\\
\text{subject to} \quad & -w_{lk} - \varphi_{lk} - \varphi'_{lk} \leq 2f_{lk}, && (4.45g)\\
& w_{lk} = \varphi_{lk} = 0, && (4.45h)\\
& \varphi'_{lk} \geq 0, && (4.45i)\\
& \text{At least one of (4.45g) or (4.45i) is tight,} && (4.45j)
\end{aligned}$$

where $\mathcal{F}_k$ and the values of the dual variables $\{\varphi_{kl} | \{k,l\} \in \mathcal{F}_k, k < l\}$ and $\{\varphi'_{kl} | \{k,l\} \in \mathcal{F}_k, l < k\}$ are to be determined. From the discussion on Figure 4.2 in Section 4.1.3, recall that only the dual variable associated with a constraint $y_k - x_{kl} \geq 0$ may assume a positive value, and the rest of the constraints in $\Delta(\{k,l\})$ belong to the set $\Delta_0(\{k,l\})$. These restrictions for $\{k,l\} \in \mathcal{F}_k$ are imposed by the constraints (4.45c)-(4.45d) and (4.45h)-(4.45i) for $k < l$ and $l < k$, respectively. The dual feasibility of the $x-$variables in the new linking constraints is mandated by the constraints (4.45b) and (4.45g), and constraints (4.45e) and (4.45j) are the counterparts of (4.11e).

For $\{k,l\} \in \mathcal{F}_k, k < l$, the optimal solution of (4.45a)-(4.45e) is identified below, and the case for $l < k$ can be derived analogously.

1. If $f_{kl} < 0$, the optimal value of $\varphi_{kl} = -2f_{kl}$ since φ_{kl} is a nonnegative variable

that appears with a negative coefficient in (4.45b) and with a positive coefficient in the objective (4.45a). In this case, x_{kl} is the basic variable associated with the new linking constraint $y_k - x_{kl} \geq 0$. For the other two linking constraints in $\Delta(\{k,l\})$, the associated surplus variables are selected as basic.

2. If $f_{kl} \geq 0$, the optimal value of $\varphi_{kl} = 0$, and all new basic variables are surplus variables.

Thus, we conclude that $\alpha_{kl} = \max(-2f_{kl}, 0)$, where $2f_{kl}$ is the objective function coefficient of x_{kl} in (3.6)-(3.12). Based on this very simple structure of the optimal solutions of (4.45a)-(4.45e) and (4.45f)-(4.45j), we re-state the row-generating PSP for QSC:

$$\zeta_{yx} = \min_{k \in K \setminus \bar{K}} (f_{kk} - \sum_{j \in J} A_{jk} u_j - \max_{\mathcal{F}_k \in \mathcal{P}_k} (\sum_{\{k,l\} \in \mathcal{F}_k, k<l} \max(0, -2f_{kl}) + \sum_{\{l,k\} \in \mathcal{F}_k, l<k} \max(0, -2f_{lk}))). \quad (4.46)$$

If $\zeta_{yx} = \bar{c}_k < 0$ with an associated optimal family of index sets $\mathcal{F}_k$, then SRMP grows both horizontally and vertically with the addition of the variables $\{y_l | l \in \Sigma_k\}$, $\{x_n | n \in S_N(\Sigma_k)\}$, and the linking constraints $\Delta(\Sigma_k)$, where $\Sigma_k = \cup_{\{k,l\} \in \mathcal{F}_k} \{k,l\}$. Furthermore, as explained at the beginning of this section, all relevant linking constraints associated with the variables $\{y_l | l \in (\Sigma_k \setminus \{k\})\}$ are also incorporated into the SRMP. Note that some of the variables $\{y_l | l \in \Sigma_k\}$ are already a part of the SRMP. This is generally true for CDR-problems with interaction.

EXAMPLE 4.2.1 *Consider the following symmetric positive semidefinite cost matrix* $\mathbf{F}$ *and the cover matrix* $\mathbf{A}$*:*

$$\mathbf{F} = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix}, \mathbf{A} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

Solving the QSC instance defined by $\mathbf{F}$ *and* $\mathbf{A}$ *by excluding the possibility of adding several* $y-$*variables simultaneously to the SRMP would result in the following PSP:*

$$\zeta_{yx} = \min_{k \in K \setminus \bar{K}} (f_{kk} - \sum_{j \in J} A_{jk} u_j - \sum_{l \in \bar{K}, k<l} \max(0, -2f_{kl}) - \sum_{l \in \bar{K}, l<k} \max(0, -2f_{lk})). \quad (4.47)$$

Suppose that we form the initial SRMP with y_1 *only. The PSP defined in* (4.47)*, which ignores the minimal variable sets of the form* $\{y_k, y_l\}, l \notin \bar{K}$ *for pricing out* $y_k, k \notin \bar{K}$*, cannot identify any new* $y-$*variable with a negative reduced cost, and the column generation terminates with* $y_1 = 1$ *and an objective function value of 2. Taking also into account the minimal variable sets* $\{y_k, y_l\}, l \notin \bar{K}$ *by solving the row-generating PSP* (4.46) *yields the optimal families of index sets* $\mathcal{F}_2 = \{\{1,2\},\{2,3\},\{2,4\}\}$*,* $\mathcal{F}_3 = \{\{1,3\},\{2,3\},\{3,4\}\}$*, and* $\mathcal{F}_4 = \{\{1,4\},\{2,4\},\{3,4\}\}$ *for* y_2, y_3*, and* y_4*, respectively. The associated reduced costs* $\bar{c}_2, \bar{c}_3, \bar{c}_4 < 0$*, and in all cases all remaining* $y-$*variables are incorporated into the SRMP. The resulting true optimal solution of* (3.6)-(3.12) *is given by* $y_1 = 0$*,* $y_2 = y_3 = y_4 = 1$ *with an objective value of 1.*

Although our aim is to find a variable with a negative reduced cost, the optimal solution of the row-generating PSP may require the addition of many columns simultaneously. Let us denote the cardinality of a set by the operator $|.|$. Then, the addition of a $y-$variable introduces $3|\bar{K}|$ many linking constraints and $|\bar{K}|$ many $x-$variables. The subproblem (4.46) adds every minimal variable set (k, l) for which $f_{kl} < 0$. Clearly, this increases the size of SRMP considerably. Moreover, even after this effort, some of the variables in these minimal variable sets may never enter the basis. Hence, we will present an algorithm that searches for a family of index sets $\mathcal{F}_k$ that results in a negative reduced cost column y_k and, more importantly, introduces the smallest set of variables $\{y_l | l \in (\Sigma_k \cap K \backslash \bar{K})\}$ to the SRMP.

We propose the approach detailed in Algorithm 2. As we mentioned above, our first aim is to find a family of minimal variable sets that makes the reduced cost of a variable negative. Our second aim is to minimize the number of variables to be added to the SRMP. Prior to this algorithm, we can solve the subproblem (4.47) which considers adding columns one-by-one. If there is no column with negative reduced cost in $K \backslash \bar{K}$, then we move to Algorithm 2. Therefore, when we start Algorithm 2, we already have the reduced costs of the variables $y_k, k \in K \backslash \bar{K}$ (line 1, Algorithm 2). We define an array z_k that contains the elements $f_{kl} < 0$, where $l \in K \backslash \bar{K}$ in the kth row of the cost matrix F (line 5, Algorithm 2). We then sort the elements in z_k in ascending order to minimize the number of variables that are needed to make the reduced cost of y_k negative (line 7, Algorithm 2). Then, we start adding the elements of z_k to $\bar{c}_k$ one-by-one (line 9, Algorithm 2). If $\bar{c}_k$ becomes negative, then we add the indices corresponding to the values in z_k that are used to make y_k a negative reduced cost column to N_k (line 11, Algorithm 2). When all $k \in K \setminus \bar{K}$ are processed, the column generation is terminated, if there is no negative reduced cost column. Otherwise, we select the minimum cardinality set N_k with $\bar{c}_k < 0$ (line 17, Algorithm 2) and then, add it to SRMP.

Algorithm 2: `Algorithm to solve the row-generating PSP for QSC.`

1 **Input**: $\bar{c}_l$ for each $l \notin \bar{K}$ computed according to (4.47)
2 **for** $k \in K \setminus \bar{K}$ **do**
3   **for** $l \in K \setminus \bar{K}$ **do**
4   **if** $f_{kl} < 0$ **then**
5   Concatenate $2f_{kl}$ to z_k;
6
7   Sort z_k in ascending order;
8   **for** t=1 **to** $|z_k|$ **do**
9         $\bar{c}_k = \bar{c}_k + z_{kt}$;
10   **if** $\bar{c}_k < 0$ **then**
11             $N_k \leftarrow order(1\, to\, t)$;
12   Go to line 2;
13
14 **if** $\bar{c}_k > 0\ \forall k \in K \setminus \bar{K}$ **then**
15   Terminate column generation ;
16 **else**
17     $u = \arg\min_{k \in K \setminus \bar{K}, \bar{c}_k < 0} |N_k|$;
18     $\zeta_{yx} = \bar{c}_u$;
19

We next generate a set of instances for the QSC problem and solve it using the column-and-row generation algorithm given in Algorithm 2. To generate the set of instances, we determine five values for the number of columns $|K| \in \{100, 150, 200, 250, 300\}$, and three values for the number of covering constraints $|J| \in \{50, 75, 100\}$. The number of $x-$variables for an instance is $\frac{(|K|-1)|K|}{2}$ and the number of linking constraints is $3\frac{(|K|-1)|K|}{2}$. For a column $k \in K$, the coefficient $A_{jk} = 1$, if a random number $U(0, 1)$

Table 4.3: Comparison of algorithms on QSC test instances.

Instances		**CPLEX**		**CRG**	
# Cov. Const.	# $y-$var.	OFV	Time	# $y-$var. gen.	Time
50	100	9,082.77	2.13	80	1.11
50	150	7,860.86	7.16	121	2.59
50	200	7,332.89	28.88	169	5.72
50	250	7,258.90	111.05	230	20.89
50	300	6,959.41	299.78	294	86.45
75	100	10,185.20	2.11	78	1.47
75	150	8,882.87	8.86	116	3.25
75	200	8,277.32	40.48	187	10.34
75	250	8,233.85	167.56	237	42.59
75	300	8,225.77	415.33	284	95.73
100	100	10,337.50	2.06	80	1.89
100	150	9,967.91	10.83	111	4.67
100	200	9,357.93	56.25	174	10.20
100	250	9,171.65	212.20	235	38.05
100	300	8,365.23	619.02	283	116.94
Average		8,633.34	132.25	178.60	29.46

is lower than 0.1. Otherwise, $A_{jk} = 0$. The diagonal elements in F are obtained by adding 1000 to a random number uniformly distributed between 0 and 300. That is, $f_{kk} = 1000 + U(0, 300)$. Using the same notation, the remaining elements are set to $f_{kl} = f_{lk} = 5 + U(-10, 0)$, $\forall k, l \in K$. The positive semi-definiteness of the matrix F is then checked for each instance. To initialize the column-and-row generation algorithm, we run the approximation algorithm of Chavatal [16], which is developed for the conventional set-covering problem and use the resulting feasible solution to form the SRMP.

Table 4.3 gives the results of the experiments. The instances are characterized by the number of covering constraints (# Cov. Const.) and the number of $y-$variables (# $y-$var.). We report the objective function value (OFV), the total number of $y-$variables (# $y-$var. gen.) generated by our column-and-row generation algorithm denoted by CRG in Table 4.3 and the solution time (Time). Our proposed algorithm outperforms CPLEX for the given instance set. The average solution time of CPLEX given at the bottom of Table 4.3 is approximately four times that of our column-and-row generation algorithm.

4.2.3 Time-Constrained Routing Problem

In [2], a time-constrained routing (TCR) problem motivated by an application that needs to schedule the visit of a tourist to a given geographical area as efficiently as possible in order to maximize his/her total satisfaction is studied. The goal is to send the tourist to one tour on each day during the vacation period while ensuring that each attraction is visited no more than once.

TCR is formulated as a set packing problem with side constraints. To be consistent with (MP), we change the notation given in [2]. The set of sites that may be visited by a tourist in a vacation period M is denoted by J, and K represents the set of daily tours that originate from and terminate at the same location. Each tour is a sequence of sites to be visited on the same day, provided that it satisfies the time-windows restrictions of the tourist and the other feasibility criteria. The total satisfaction of the tourist from participating in tour k is given by c_k, and the binary variable y_k is set to one, if tour k is incorporated into the itinerary of the tourist. If tour k is performed on day m, the binary variable x_{km} takes the value one. The overall mathematical model is given as

$$
\begin{aligned}
\text{maximize} \quad & \sum_{k \in K} c_k y_k, && && (4.48)\\
\text{subject to} \quad & \sum_{k \in K} A_{jk} y_k \leq 1, && j \in J, && (4.49)\\
& \sum_{k \in K} B_{km} x_{km} = 1, && m \in M, && (4.50)\\
& y_k - \sum_{m \in M} D_{km} x_{km} = 0, && k \in K, && (4.51)\\
& y_k \in \{0,1\}, && k \in K, && (4.52)\\
& x_{km} \in \{0,1\}, && k \in K, m \in M, && (4.53)
\end{aligned}
$$

where $A_{jk} = 1$, if tour k contains the site j, $A_{jk} = 0$ otherwise; $B_{km} = D_{km} = 1$, if tour k can be performed on day m, $B_{km} = D_{km} = 0$ otherwise. By constraints (4.49), at most one tour in the selected itinerary of the tourist visits site j. Constraints (4.51) impose that a tour to be included in the itinerary is assigned to one of the days allowed in M, and we also require that exactly one tour is selected on each day of the vacation period as prescribed by constraints (4.50). Finally, the objective (4.48) maximizes the total satisfaction of the tourist over the vacation period M.

In [2], TCR is solved heuristically in two steps. In the first step, the LP relaxation of (4.48)-(4.53) is solved by a column-and-row generation approach due to a potentially huge number of tours. To this end, a large number of possible tours is enumerated and added to a set K. A subset $\bar{K} \subset K$ of these tours is selected to form the SRMP for the column-and-row generation procedure. At each iteration, a set of new tours $k \in L \subseteq (K \setminus \bar{K})$ is introduced to the SRMP. For each $k \in L$, this implies adding x_{km} for all m such that $B_{km} = 1$, and the associated linking constraint $y_k - \sum_{m \in M} D_{km} x_{km} = 0$ to the SRMP. The row-and-column generation procedure terminates when the condition stated below in Theorem 4.2.1 is satisfied. Observe that the authors evaluate this condition for each tour in $k \in (K \setminus \bar{K})$ following each optimization of the SRMP, where K is known explicitly. Furthermore, in the computational experiments $\mid L \mid = 1$, that is, the tour that violates the condition in Theorem 4.2.1 to the largest extent is added to the SRMP. In the second step of the proposed solution approach, (4.48)-(4.53) is solved by a commercial solver over the set of tours currently present in the SRMP in order to obtain an integer feasible solution for TCR.

Now, let $\{u_j | j \in J\}$, $\{v_m | m \in M\}$, and $\{w_k | k \in K\}$, denote the dual variables associated with the constraints (4.49)-(4.51). The following theorem given in [2] without a proof defines the stopping condition for the column-and-row generation algorithm of the authors:

Claim 4.2.1 (adapted from [2]) *The solution of the current RMP is optimal for the LP relaxation of* (4.48)-(4.53) *if* $\bar{c}_k = c_k - \sum_{j \in J} A_{jk} u_j - \sum_{m \in M} B_{km} v_m \leq 0$, *for each* $k \in K$.

The statement of the claim above corrects two typos in the original Theorem 3.1 in [2], where the termination condition appears as $\bar{c}_k = c_k - \sum_{j \in J} A_{jk} u_j - \sum_{m \in M} B_{km} v_m \geq 0$, for each $m \in M$ and k such that $B_{km} = 1$. Next, we demonstrate that the stopping condition in Claim 4.2.1 is incorrect and may lead to a suboptimal solution when the column-and-row generation algorithm proposed in [2] terminates. Then, we discuss how the generic column-and-row generation algorithm proposed in this chapter may be applied here in order to solve the proposed large-scale LP correctly.

The following is the dual of the linear programming relaxation of (4.48)-(4.53):

$$\begin{aligned}
\text{minimize} \quad & \sum_{j \in J} u_j + \sum_{m \in M} v_m, && && (4.54)\\
\text{subject to} \quad & \sum_{j \in J} A_{jk} u_j + w_k \geq c_k, && k \in K, && (4.55)\\
& -w_k + v_m \geq 0, && k \in K, m \in \{m \in M : B_{km} = 1\}, && (4.56)\\
& u_j \geq 0, && j \in J. && (4.57)
\end{aligned}$$

Given the optimal dual variables associated with the current SRMP, the resulting pricing subproblem to be solved is stated as

$$\zeta_{yx} = \max_{k \in K \setminus \bar{K}} \quad \bar{c}_k, \tag{4.58}$$

where $\bar{c}_k = c_k - \sum_{j \in J} A_{jk} u_j - w_k$ denotes the reduced cost of tour k. If the optimal objective function value of this subproblem is positive with $\bar{c}_k > 0$, the variables y_k, x_{km} for all $m \in M$ such that $B_{km} = 1$, and the primal constraint $y_k - \sum_{m \in M} D_{km} x_{km} = 0$ should be added to the SRMP. Otherwise, the optimal solution of the current SRMP is declared as optimal for the LP relaxation of (4.48)-(4.53), and the algorithm terminates. The challenge here is that the value of the dual variable w_k is unknown because the corresponding constraint is currently absent from the SRMP. Hence, in order to design an optimal column-and-row generation algorithm for TCR, one has to devise a method that anticipates the correct value of w_k to be incorporated into the pricing subproblem. In the sequel, we first show that the condition in Claim 4.2.1 fails to do so and then illustrate how this is accomplished based on the generic column-and-row generation framework proposed in this chapter.

Consider an iteration of the column-and-row-generation algorithm, where the current SRMP is solved to optimality and the associated optimal dual solution is represented by $u_j, j \in J$, $w_k, k \in \bar{K}$, and $v_m, m \in M$. Suppose that $y_{k'}, k' \in K \setminus \bar{K}$ is to be priced out, where $\bar{c}_{k'} = c_{k'} - \sum_{j \in J} A_{jk} u_j - \sum_{m \in M} B_{km} v_m \leq 0$. Observe for the currently unknown dual variable $w_{k'}$ that the dual constraint set (4.56) implies that

$$w_{k'} \leq \min_{m \in M: B_{k'm}=1} v_m, \tag{4.59}$$

and if $| \{m \in M : B_{k'm} = 1\} | > 1$ and $\max_{m \in M: B_{k'm}=1} v_m > 0$, then we may have

$$w_{k'} \leq \min_{m \in M: B_{k'm}=1} v_m < \sum_{m \in M: B_{k'm}=1} v_m. \tag{4.60}$$

Clearly, this may lead to

$$\bar{c}_{k'} = c_{k'} - \sum_{j \in J} A_{jk'} u_j - \sum_{m \in M} B_{k'm} v_m \leq 0 < c_{k'} - \sum_{j \in J} A_{jk'} u_j - w_{k'} = \bar{c}_{k'}. \tag{4.61}$$

Thus, we conclude that while $\bar{c}_k \leq 0$ for all $k \in K$ as required by Theorem 4.2.1 due to Avella et al. [2], there may exist a tour k' with $\bar{c}_{k'} > 0$. In other words, $\sum_{m \in M} B_{km} v_m$ is not an appropriate estimate of the missing dual variable $w_{k'}$, and the column-and-row generation algorithm given in [2] may terminate prematurely while there exists a tour k' with a positive reduced cost. On the other hand, if $v_m \leq 0$ for all $m \in M$ such that $B_{k'm} = 1$,

$$\sum_{m \in M: B_{k'm}=1} v_m < w_{k'} \leq \min_{m \in M: B_{k'm}=1} v_m \tag{4.62}$$

may hold, and this may result in

$$\bar{c}_{k'} = c_{k'} - \sum_{j \in J} A_{jk'} u_j - w_{k'} < 0 < c_{k'} - \sum_{j \in J} A_{jk'} u_j - \sum_{m \in M} B_{k'm} v_m = \bar{c}_{k'}. \tag{4.63}$$

That is, a tour k' with a negative reduced cost may violate the condition in Claim 4.2.1 and is added to the SRMP unnecessarily.

Using our proposed column-and-row generation algorithm, the pricing subproblem becomes a two-stage problem given by

$$\zeta_{yx} = \max_{k \in K \setminus \bar{K}} c_k - \sum_{j \in J} A_{jk} u_j - \alpha_k \tag{4.64}$$

$$\alpha_k = \text{minimize} \quad w_k \tag{4.65}$$

$$\text{subject to} \quad w_k \leq v_m, \quad m \in \{m \in M : B_{km} = 1\}, \tag{4.66}$$

$$\text{at least one constraint in (4.66) is tight.} \tag{4.67}$$

To calculate the correct reduced cost of y_k, we solve the problem (4.65)-(4.67) separately for each $k \in K \setminus \bar{K}$. Since TCR problem is a CDR-problem with no interaction, $\Delta_0(k) = \emptyset$ for any $k \in K \setminus \bar{K}$ and $\Sigma_k = \{k\}$. Moreover, $y_k - \sum_{m \in M} D_{km} x_{km} = 0$ belongs to $\Delta_+(k)$. The constraint set (4.66) corresponds to (4.11b) and imposes that the dual constraints (4.56) corresponding to the variables x_{km} for all $m \in M$ such that $B_{km} = 1$ induced by any $k \in K \setminus \bar{K}$ are not violated. The constraint (4.67) corresponds to (4.11e) and imposes the complementary slackness condition. Consequently, if the optimal objective of (4.64) is $\zeta_{yx} = \bar{c}_k > 0$, then the SRMP is augmented with y_k, x_{km} for all $m \in M$ such that $B_{km} = 1$, and one new linking constraint of type (4.50), and y_k is the natural candidate to enter the basis during the next iteration of the column-and-row generation scheme. In order to warm-start the primal simplex algorithm to re-optimize the SRMP, the optimal basis from the previous iteration must be augmented with a new variable. This new basic variable x_{km} is provided by the optimal solution of (4.65)-(4.67) for y_k and corresponds to a tight constraint in (4.66).

For each $k \in K \setminus \bar{K}$, the solution of (4.65)-(4.67) is trivial and $\alpha_k = w_k = \min_{m \in M : B_{km} = 1} v_m$. We next formulate the correct termination criterion for a column-and-row generation algorithm for TCR in Theorem 4.2.1. The proof of this theorem follows from the analysis of the column-and-row generation algorithm given in Section 4.1. We then conclude with a small numerical example for which the stopping condition of [2] as stated in Claim 4.2.1 would lead to a premature termination of the column-and-row generation algorithm with a suboptimal solution. This example and a Lagrangian way to find a correct termination criterion are given in [60].

THEOREM 4.2.1 *The solution of the current SRMP is optimal for the LP relaxation of* (4.48)-(4.53) *if* $c_k - \sum_{j \in J} A_{jk} u_j - \min_{m \in M: B_{km}=1} v_m \leq 0$ *for every* $k \in K \setminus \bar{K}$, *for each* $k \in K$.

EXAMPLE 4.2.2 *Consider an instance of TCR with 3 sites, 4 tours, 2 time periods, and assume that the SRMP is initialized with the first 3 three tours. All other data are specified in Table 4.4, where the optimal dual solution of the initial SRMP is provided in Columns "*u_j^0*," "*w_k^0*," and "*v_m^0*." The values of the non-zero variables in the corresponding optimal primal solution are* $y_1^0 = 1, x_{11}^0 = 1, y_2^0 = 0, y_3^0 = 1, x_{32}^0 = 1$ *with an objective function value of 7.*

Table 4.4: Counterexample for the optimality of the algorithm proposed by [2].

Site j	$\{k \mid A_{jk} = 1\}$	u_j^0	u_j^1	**Tour** k	c_k	$\{m \mid B_{km} = 1\}$	w_k^0	w_k^1	**Period** m	v_m^0	v_m^1
1	{1, 4}	0	1	1	5	{1}	3	3	1	3	3
2	{1, 2}	2	1	2	4	{1, 2}	2	3	2	2	3
3	{3, 4}	0	0	3	2	{2}	2	2			
				4	4	{1, 2}		3			

Following Claim 4.2.1, the reduced cost of y_4 is computed as

$$\bar{c}_4 = c_4 - \sum_{j \in J} A_{j4} u_j^0 - \sum_{m \in M} B_{4m} v_m^0 = r_4 - (u_1^0 + u_3^0) - (v_1^0 + v_2^0) = -1,$$

and the column-and-row generation algorithm terminates. However, the correct reduced cost of y_4 is given by

$$\bar{c}_4 = c_4 - \sum_{j \in J} A_{j4} u_j^0 - \min_{m \in M: B_{4m}=1} v_m^0 = c_4 - (u_1^0 + u_3^0) - \min(v_1^0, v_2^0) = 2$$

according to Theorem 4.2.1, and the column-and-row generation algorithm proceeds after augmenting the SRMP with y_4. In the next iteration, re-solving the SRMP provides us with an optimal solution $y_1^1 = 0, y_2^1 = 1, x_{21}^1 = 1, y_3^1 = 0, y_4^1 = 1, x_{42}^1 = 1$, and an associated objective value of 8. The corresponding optimal dual solution is displayed in columns titled "u_j^1," "w_k^1," and "v_m^1" in Table 4.4.

4.3 Mixed Column-Dependent-Rows Problems

Keeping the discussion on the CDR-problems with and with no interaction separate helped us to highlight the differences in developing the row-generating PSP for these two types of problems in the past sections. These tools of analysis may however also be combined to tackle CDR-problems in which some minimal variable sets are of cardinality one, while others are composed of two or more $y-$variables. We aptly refer to such problems as *mixed* CDR-problems. To the best of our knowledge, there is no study in the literature that deals with a mixed CDR-problem. Therefore, to complement our discussion with such an extension, we briefly describe a fictitious mixed CDR-problem in the subsequent part of this section.

Consider a tactical network design and vehicle routing problem defined on a directed network, where the set of nodes is represented by K. A demand b_k is associated with each node of the network, and these demands have to be served daily by a set of routes originating and terminating at a depot, possibly located at two different nodes of the network. We pay a fixed cost of c_k for opening a depot at a node $k \in K$, and each vehicle route incurs a cost of d_n. Assume that split deliveries are allowed, where the number of units delivered to customer k by route n is given by B_{kn}. The objective is to minimize the total fixed and routing costs. No more than v_k vehicles may be dispatched from a depot at node k to return to the same location. Such routes are referred to as tours. Similarly, the number of vehicles which originate at node k and terminate at node l cannot exceed v_{kl}. The set of tours starting at node k is denoted by N_k, and N_{kl} represents the set of routes emanating from node k and finishing at node l. The set of all routes is given by $N = (\cup_{k \in K} N_k) \cup (\cup_{k,l \in K, k \neq l} N_{kl})$. The variable $y_k, k \in K$, takes the value one, if a depot is located at node k, and is zero otherwise. The binary variable $x_n, n \in N$, indicates whether tour/route n is selected in the solution. Then, the LP relaxation of this problem may be formulated as below:

$$\text{minimize} \quad \sum_{k\in K} c_k y_k + \sum_{n\in N} d_n x_n, \tag{4.68}$$

$$\text{subject to} \quad \sum_{n\in N} B_{kn} x_n \geq b_k, \quad k \in K, \tag{4.69}$$

$$v_k y_k - \sum_{n\in N_k} x_n \geq 0, \quad k \in K, \tag{4.70}$$

$$v_{kl} y_k - \sum_{n\in N_{kl}} x_n \geq 0, \quad k,l \in K,\ k \neq l, \tag{4.71}$$

$$v_{kl} y_l - \sum_{n\in N_{kl}} x_n \geq 0, \quad k,l \in K,\ k \neq l, \tag{4.72}$$

$$0 \leq y_k \leq 1, k \in K, \quad 0 \leq x_n \leq 1, n \in N. \tag{4.73}$$

The objective (4.68) minimizes the total cost of opening depots and serving the customers by a set of routes. Constraints (4.69) ensure that the customer demands are satisfied. The set of linking constraints (4.70) prescribe that a tour associated with $k \in K$ is not selected unless a depot is located at node k. Similarly, a route from node k to node l requires that a depot is present at both nodes k and l as described by the set of linking constraints (4.71)-(4.72).

This problem is a special case of the generic model (MP), where A, a, and r are zero, and $M = K$. In general, the number of routes in the problem may grow exponentially which motivates a column generation algorithm for solving the formulation (4.68)-(4.73). If, in addition, the number of nodes in the network is large, then this would merit a column-and-row generation approach because including all $\mathcal{O}(\mid K \mid^2)$ linking constraints in the formulation (4.68)-(4.73) directly would be computationally prohibitive. The column-and-row generation would be initialized with a small number of depots located at nodes $k \in \bar{K}$ and a set of associated routes so that the initial SRMP is feasible. For this problem, no $y-$PSP is required because the generation of a new $y-$variable is only meaningful if associated $x-$variables and linking constraints are introduced into the SRMP along with it. In the $x-$PSP, we either construct a tour associated with a depot at a node $k \in \bar{K}$ or a route from a depot at a node $k \in \bar{K}$

to a depot at a node $l \in \bar{K}$. In the row-generating PSP for y_k, $\{y_k\}$ is a minimal variable set of cardinality one for a single linking constraint (4.70). The remaining minimal variable sets are either of the form $\{y_k, y_l\}$ or $\{y_l, y_k\}$, depending on whether routes are generated from node k to node l or vice versa. That is, except for one, all minimal variable sets in the row-generating PSP for y_k are of cardinality two and induce two linking constraints (4.71)-(4.72). In both cases, $y-$variables generate the associated $x-$variables, and the linking constraints are redundant until all variables in the associated minimal variable set take positive values. Thus, both Assumptions 3.1.1 and 3.1.2 are satisfied. Note that there is a one-to-one correspondence between the minimal variable sets and the linking constraints. Moreover, the linking constraints are in the form mandated by (3.1) in Assumption 3.1.3. We conclude that the problem at hand is a mixed CDR-problem and is amenable to the column-and-row generation framework devised in this chapter. We only need to pay attention to set up (4.11) in the row-generating PSP properly depending on the cardinality of the associated minimal variable set.

Chapter 5

COMBINATION WITH LAGRANGIAN RELAXATION

In this chapter, we consider another approach based on combining Lagrangian relaxation (LR) and column generation (CG) to solve the CDR problems. This approach results from a different observation than that in Chapter 4 because the increase in the number of rows must now be taken into account by an increase in the number of Lagrange multipliers. More explicitly, the generation of a minimal variable set $\{y_k | k \in S_K\}$ triggers the generation of $\{x_n | n \in S_N(S_K)\}$ and a set of linking constraints $\Delta(S_K)$, which is dualized in the objective function so that a set of new multipliers are added to the set of existing Lagrange multipliers. In the next section, we give an introduction to the combination of LR and CG algorithms and present a methodology that combines these two approaches to solve the CDR problems. In the second section, we apply this algorithm to the TCR problem which belongs to the class of CDR-problems. Moreover, we conduct computational experiments for this problem on a set of randomly generated instances. The performance of this algorithm is compared with that of the simultaneous column-and-row generation algorithm defined in Chapter 4.

5.1 Proposed Solution Method

Column Generation vs. Lagrangian Relaxation. LR and CG algorithms are two prominent algorithms to solve integer programming problems. CG algorithm is used to solve Dantzig-Wolfe decompositions although this algorithm can also be applied directly to a compact formulation, e.g., as in vehicle routing problem with time-windows. The relationship between Dantzig-Wolfe decomposition and Lagrangian relaxation is significant. As we explained in Chapter 2, LR is obtained by dualizing exactly those constraints that are the linking constraints in the Dantzig-Wolfe decomposition. Moreover, the subproblems in the corresponding LR and CG algorithms are identical except for a constant in the Lagrangian subproblems. In the column generation procedure, the values for the dual variables are obtained by solving the LP relaxation of the RMP, whereas in the LR, the Lagrange multipliers can be updated by the simple subgradient optimization (see [47] for a detailed comparison of these methods).

Both approaches have advantages and disadvantages. For example, in CG, while solving the RMP by the simplex algorithm, a tailing-off effect; i.e., slow convergence towards the optimum in the final phase of the algorithm, is generally observed. In LR, the subgradient optimization method provides a fast update of the Lagrange multipliers but we usually have to stop the procedure without a proof of optimality. There exist approaches which combine these two methods to exploit the strengths of both. An overview of the possible combinations of these methods is given in [47]. They discuss two ways in which the two methods can be combined efficiently:

- LR can be applied to the master problem to approximate the optimal values of the dual variables. Two examples using this combination are [15] and [48], which solve the discrete lot-sizing and scheduling problem. LR is employed to solve the master problem in order to reduce the degeneracy.

- In the second one, LR can be used on the original formulation of the problem to generate columns since the pricing subproblem in Dantzig-Wolfe decomposition is the same as the LR subproblem except for a constant. Hence, after the pricing subproblem finds a negative reduced cost column, LR is solved by subgradient optimization and the generated columns are added to the RMP. These subgradient iterations are considerably faster than solving the RMP at each iteration. An example using this combination is given in [6], which solves the plant location problem.

It is clear that we can apply the former combination to CDR problems by dualizing the linking constraints into the objective function. For a given set of multipliers $w \geq 0$, the resulting LR problem is given by

$$\begin{aligned} Z(w) = \text{minimize} \quad & \sum_{k \in K} c_k y_k + \sum_{n \in N} d_n x_n + \\ & \sum_{i \in I} w_i (r_i - \sum_{k \in K} C_{ik} y_k - \sum_{n \in N} D_{in} x_n), \\ \text{subject to} \quad & \sum_{k \in K} A_{jk} y_k \geq a_j, \quad j \in J, \\ & \sum_{n \in N} B_{mn} x_n \geq b_m, \quad m \in M, \\ & y_k \geq 0, k \in K; \quad x_n \geq 0, n \in N. \end{aligned} \tag{5.1}$$

The dual of (5.1) then becomes

$$\begin{aligned} \text{maximize} \quad & \sum_{j \in J} a_j u_j + \sum_{m \in M} b_m v_m + \sum_{i \in I} r_i w_i, \\ \text{subject to} \quad & \sum_{j \in J} A_{jk} u_j \leq c_k - \sum_{i \in I} C_{ik} w_i, \quad k \in K, \\ & \sum_{m \in M} B_{mn} v_m \leq d_n - \sum_{i \in I} D_{in} w_i, \quad n \in N, \\ & u_j \geq 0, j \in J; \quad v_m \geq 0, m \in M, \end{aligned} \tag{5.2}$$

where $\{u_j | j \in J\}$ and $\{v_m | m \in M\}$ are the sets of dual variables corresponding to the first and the second constraint sets of (5.1), respectively, and $\{w_i | i \in I\}$ is the set of Lagrange multipliers. Suppose that the optimal objective function value of the original CDR problem is Z. The optimal objective function value of (5.1) for any $w \geq 0$ is $Z(w) \leq Z$. To find the best possible lower bound attainable through (5.1) we solve the Lagrangian dual (LD) problem

$$Z_{LD} = \max_{w \geq 0} Z(w). \tag{5.3}$$

Since the LR problem in (5.1) is a LP problem, that, the extreme points can be fractional, theoretically $Z_{LD} = Z$ (see [62] for the formal explanations of these statements).

Subgradient Optimization. Subgradient optimization algorithm is an iterative method and at each iteration problem (5.1) is solved. The values of the multipliers are initialized as w^0. Let $(y^*(w^t), x^*(w^t))$ be the optimal, possibly infeasible, primal solution when (5.1) is solved at iteration t using the multiplier vector w^t. Additionally, let $g_i^t, i \in I$ be the subgradient vector associated with the relaxed constraints. The entries of this vector are calculated as $g_i^t = (r_i - \sum_{k \in K} C_{ik} y_k^*(w^t) - \sum_{n \in N} D_{in} x_n^*(w^t)),\ i \in I$. The values of the multipliers can be updated in a subgradient step as follows:

$$
\begin{aligned}
w_i^{t+1} &= \max(0, w_i^t + \lambda^t g_i^t), &\quad (5.4)\\
\lambda^t &= \frac{\theta(UB - Z(w^t))}{\sum_{i \in I}(g_i^t)^2}, &\quad (5.5)
\end{aligned}
$$

where λ^t is the step size, UB is the objective function value of any feasible solution to the problem. The parameter θ is initialized. If there is no progress (increase) in the lower bound for some given number of iterations, say S, then θ is updated as $\theta \to \theta/2$. The subgradient iterations are terminated when a given number of iterations are performed or θ is lower than a given small value. The lower bound value $Z(w^t)$ is not a nondecreasing function of t. Hence, we denote the maximum lower bound found over all subgradient iterations as Z_{max}. It is updated at each subgradient iteration as $Z_{max} = \max(Z_{max}, Z(w^t))$. At the end of the subgradient phase, the Lagrange multipliers and Z_{max} are approximations of the optimal dual variables and Z_{LD}, respectively. Using these multipliers, problem (5.1) is solved again. We refer to [10, 43, 44] for details about the implementation of subgradient optimization.

Solution Approach. In this section, we give the details of the combination of CG and LR algorithms, denoted by CG-LR, to solve the CDR problems. If the number of variables in (5.1) is large, we can solve this problem by column generation by replacing the sets K, N by their subsets $\bar{K}$ and $\bar{N}$, respectively, to form the RMP. Note that the set of constraints in this model is known in contrast to the SRMP given in Chapter 3, where the new rows are added to the model on the fly. Hence, we no longer call this problem SRMP, but solely RMP. Additionally, the notation SRMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ defined in Chapter 4 now reduces to RMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ where the last set is the set of existing multipliers residing in the objective function. There is a set of multipliers $w_i, i \in I \backslash I(\bar{K}, \bar{N})$ which does not exist in the objective function since the associated constraints have not been generated. The resulting LR problem at iteration T of the column generation algorithm is given by

$$
\begin{aligned}
Z^T(w) = \text{minimize} \quad & \sum_{k \in \bar{K}} c_k y_k + \sum_{n \in \bar{N}} d_n x_n + \sum_{i \in I(\bar{K}, \bar{N})} w_i (r_i - \sum_{k \in \bar{K}} C_{ik} y_k - \sum_{n \in \bar{N}} D_{in} x_n), \\
\text{subject to} \quad & \sum_{k \in \bar{K}} A_{jk} y_k \geq a_j, \quad j \in J, \\
& \sum_{n \in \bar{N}} B_{mn} x_n \geq b_m, \quad m \in M, \\
& y_k \geq 0, k \in \bar{K}; \quad x_n \geq 0, n \in \bar{N}.
\end{aligned}
$$

Reorganizing the terms in the model above, the LR problem becomes

$$
\begin{aligned}
Z^T(w) = \text{minimize} \quad & \sum_{k \in \bar{K}} (c_k - \sum_{i \in I(\bar{K}, \bar{N})} C_{ik} w_i) y_k + \sum_{n \in \bar{N}} (d_n - \sum_{i \in I(\bar{K}, \bar{N})} D_{in} w_i) x_n + \\
& \sum_{i \in I(\bar{K}, \bar{N})} w_i r_i, && (5.6) \\
\text{subject to} \quad & \sum_{k \in \bar{K}} A_{jk} y_k \geq a_j, \quad j \in J, && (5.7) \\
& \sum_{n \in \bar{N}} B_{mn} x_n \geq b_m, \quad m \in M, && (5.8) \\
& y_k \geq 0, k \in \bar{K}; \quad x_n \geq 0, n \in \bar{N}. && (5.9)
\end{aligned}
$$

This problem decomposes into separate problems for y and x. The resulting Lagrangian dual problem is

$$
Z_{LD}^T = \max_{w_i \geq 0, i \in I(\bar{K}, \bar{N})} Z^T(w). \tag{5.10}
$$

which is solved by the subgradient optimization method. At each iteration of the subgradient optimization algorithm, the LP problem (5.6)-(5.9) is solved. Using the values of the dual variables $\{u_j|j \in J\}$, $\{v_m|m \in M\}$, and multipliers $\{w_i|i \in I(\bar{K}, \bar{N})\}$ obtained by the subgradient optimization, we solve one of the PSPs, which are $y-$, $x-$PSPs and row generating PSP, to detect the negative reduced cost variables. The mechanism of the CG-LR algorithm is exactly like that given in Figure 4.1 in Chapter 4. Additionally, the definitions of the PSPs are similar to those given in Chapter 4 since the dual constraints in (5.2) are the same as (DMP-y) and (DMP-x). However, the values of the dual variables $\{u_j|j \in J\}$, $\{v_m|m \in M\}$, and multipliers $\{w_i|i \in I(\bar{K}, \bar{N})\}$ to check the feasibility in the dual constraints in (5.2) are provided by the subgradient optimization instead of the simplex method. Moreover, the newly added rows resulting from the row-generating PSP are dualized in the objective function so that we add a set of multipliers to the existing multipliers. More explicitly, the generation of a minimal variable set $\{y_k|k \in S_K\}$ triggers the generation of $\{x_n|n \in S_N(S_K)\}$ and a set of linking constraints $\Delta(S_K)$ which is dualized in the objective function so that a set of new multipliers $\{w_i|i \in \Delta(S_K)\}$ is added to the set of existing Lagrange multipliers. This chapter provides a different perspective for the unknown dual variables defined in Chapter 4. Later, we explain the row-generating PSP from this perspective.

Dualizing a new constraint is also used in relax-and-cut where Lagrangian bounds are attempted to be improved by dynamically strengthening relaxations with the introduction of valid inequalities (see [41, 52, 56] for the details of the relax-and cut algorithm). This algorithm avoids dualizing the set of all constraints (cuts) in exponential size to the objective function but adds them dynamically by utilizing a cut pool strategy. In relax-and-cut, the added constraints are violated constraints and are detected by solving a separation problem but in our case, the constraints to be added are not violated and they must be considered when solving the row-generating PSP.

Finally, we elaborate on one more issue related to the upper-bounding before we move on to the row-generating PSP. Recall that Z is the optimal objective function value of (MP). Suppose that LR is applied to solve the CDR problem defined as (MP). To solve the Lagrangian dual problem by subgradient optimization, we use UB, which can be obtained by using a heuristic, as the objective function value of a primal feasible solution to (MP). UB is valid throughout the algorithm since at each iteration of the subgradient optimization algorithm, the LR problem, which provides a lower-bound on Z, is solved. However, when (MP) is solved by CG-LR, at some iteration of the column generation, say T, the value of Z_{LD}^T is an approximation to the optimal objective function value of RMP, $Z_{LD}^T \approx Z^T$. Moreover, we know that at the same iteration T, the objective function value is also an upper bound on Z such that $Z^T \geq Z$. Hence, if $Z^T > UB$, then UB is not a valid upper bound on the optimal objective function of the RMP. This situation can be seen in Figure 5.1. The subgradient optimization cannot reach Z_{LD}^T since the step length λ defined in (5.4) approaches to zero when $Z^T(w) \approx UB$. Moreover, the multipliers are misguided at the subgradient update given in (5.4) since the step size λ is negative when $Z^T(w) > UB$. To overcome this issue, we use the objective function value of the previous Lagrangian dual problem Z_{LD}^{T-1} as the upper-bound for iteration T of column generation as long as $UB < Z_{LD}^{T-1}$. When $UB \geq Z_{LD}^{T-1}$, UB is valid and we use it as the upper bound because using Z_{LD}^{T-1} as the upper bound at the latest iterations of column generation causes slow convergence, as observed in computational experiments.

The case in Figure 5.1 is also observed in [48]. At various steps during the column generation process, they construct a good feasible integer solution starting from the current solution of the RMP. This upper-bound is valid since it is obtained through a heuristic considering only the existing variables in the RMP.

Figure 5.1: Bounding in the CG-LR Algorithm.

Before invoking the row-generating PSP, neither y- nor x-PSP results in a negative reduced cost column using the set of constraints in RMP and the set of multipliers $w_i, i \in I(\bar{K}, \bar{N})$ in the objective function of RMP. However, there may be some minimal variable sets not in RMP which induce a set of $x-$variables and a set of linking constraints. There are Lagrange multipliers associated with these linking constraints in the objective function which are not generated yet.

Remark 5.1.1 *Assumption 3.1.2 given in Chapter 3 implies that a feasible solution of RMP does not violate any missing linking constraint ($i \in I\backslash I(\bar{K},\bar{N})$) before all variables in at least one of the associated minimal variable sets are added to the RMP. If we initialize the multipliers with value zero, a multiplier remains at zero after a subgradient iteration as long as the associated constraint is not violated. However, $g_i^t > 0$ may affect the calculation of λ^t. As in [10], we choose to set $g_i^t = 0$ whenever $g_i^t > 0$ and $w_i = 0$.*

The row-generating PSP, as explained in Chapter 3, generates a family of the index sets of the form S_K^k, denoted by $\mathcal{F}_k$. The generation of $\mathcal{F}_k$ prompts the generation of a set of $y-$ variables indexed by Σ_k, a set of $x-$variables indexed by $S_N(\Sigma_k)$ and a set of linking constraints $\Delta(\Sigma_k)$ which is dualized in the objective function with multipliers $\{w_i | i \in \Delta(\Sigma_k)\}$. Also, the reduced costs $\bar{c}_k$ and $\bar{d}_n$ for y_k and $x_n, n \in S_N(S_K^k)$, respectively, are expressed as below for any given y_k, an associated $\mathcal{F}_k$, and $S_K^k \in \mathcal{F}_k$:

$$\bar{c}_k = c_k - \sum_{j \in J} A_{jk} u_j - \sum_{i \in I(\bar{K},\bar{N})} C_{ik} w_i - \sum_{i \in \Delta(\Sigma_k)} C_{ik} w_i, \tag{5.11}$$

$$\bar{d}_n = d_n - \sum_{m \in M} B_{mn} v_m - \sum_{i \in \Delta(S_K^k)} D_{in} w_i \tag{5.12}$$

In (5.11)-(5.12), the values of the dual variables $\{u_j | j \in J\}$, $\{v_m | m \in M\}$ and $\{w_i | i \in I(\bar{K},\bar{N})\}$, are retrieved from the subgradient optimization and due to Remark given above, $w_i = 0, i \in \Delta(\Sigma_k)$ can be selected. It may result that $\bar{c}_k > 0, k \in \Sigma_k$ and $\bar{d}_n < 0$ for some $n \in S_N(\Sigma_k)$. Hence, $\{y_l | l \in \Sigma_k\}$ and $\{x_n | n \in S_N(\Sigma_k)\}$ are added to the RMP and the linking constraints $\Delta(\Sigma_k)$ are dualized to the objective function. According to the Assumption 3.1.3, the set of linking constraints $\Delta(\Sigma_k)$ imposes that x_n cannot assume a positive value until all variables in S_K^k have a positive value. However, these linking constraints are dualized to the objective function with multipliers so that x_n can take positive values and some of the constraints in $\Delta(\Sigma_k)$ are violated. Hence, the values of the associated multipliers $\{w_i | i \in \Delta(\Sigma_k)\}$ are increased in the subgradient iteration. The question is whether we can we avoid such subgradient iterations by estimating the values of the new multipliers. It boils down to a look-ahead procedure as given in Chapter 4.

Suppose that for a given family of index sets $\mathcal{F}_k$ that minimizes the reduced cost of a variable y_k, we use the procedure given in Chapter 4 to determine the starting values of the new multipliers $\{w_i | i \in \Delta(S_K^k)\}$ for each $S_K^k \in \mathcal{F}_k$. This is accomplished by an implicit construction of a basic optimal solution to RMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$. Moreover, the values of the multipliers $\{w_i | i \in \Delta(S_K^k)\}$ are determined by solving the two-stage problem (4.11a)-(4.11e) defined in Chapter 4 and they satisfy the following conditions:

- The objective in (4.11a) makes sure that the minimum possible reduced cost of $y_k, k \in K \backslash \bar{K}$ is obtained.

- The constraints (4.11c)-(4.11d) impose that $w_i = 0, i \in \Delta_0(\Sigma_k)$ and $w_i \geq 0, i \in \Delta_+(S_K^k)$ where $C_{il} = 0, l \in \bar{K}$ and $D_{in} = 0, n \in \bar{N}$ for $i \in \Delta_+(S_K^k)$. Hence, the reduced costs of the variables $\{y_l | l \in \bar{K}\}$ and $\{x_n | n \in \bar{N}\}$ do not change.

- According to (4.11b), the reduced costs of $\{x_n | n \in S_N(\Sigma_k)\}$ are nonnegative. Hence, the dualization of the new linking constraint set $\Delta(\Sigma_k)$ with multipliers satisfying (4.11b)-(4.11d) results in a set of variables $\{x_n | n \in S_N(\Sigma_k)\}$ having nonnegative reduced cost added to RMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$. Therefore, the optimal values of the dual variables $\{u_j | j \in J\}$, $\{v_m | m \in M\}$, and $\{w_i | i \in I(\bar{K}, \bar{N})\}$ in RMP$(\bar{K}, \bar{N}, I(\bar{K}, \bar{N}))$ do not change and the values set for $\{w_i | i \in \Delta(\Sigma_k)\}$ in the row-generating PSP are optimal for RMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$.

- Lastly, the complementary slackness condition (4.11e) imposes that $|\Delta(S_K^k)|$ of the tight constraints in (4.11b)-(4.11d) are linearly independent. In general, the subgradient optimization algorithm does not necessarily need this condition to terminate. However, the values of the new multipliers $\{w_i | i \in \Delta(\Sigma_k)\}$ determined by (4.11a)-(4.11e) are the optimal dual variable values resulting from the solution of RMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$ as mentioned in the previous item. If RMP$(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$ is solved by subgradient optimization starting with $w_i = 0, i \in \Delta(\Sigma_k)$, then the values of the multipliers $\{w_i | i \in \Delta(\Sigma_k)\}$

would be the approximations of those found by (4.11a)-(4.11e) when the subgradient optimization terminates. Due to constraint (4.11e), the reduced cost of some $x_n, n \in S_N(\Sigma_k)$ may be zero; i.e., (4.11b) is tight for x_n. This means that there is an alternate optimal solution for $\text{RMP}(\bar{K}, \bar{N} \cup S_N(\Sigma_k), I(\bar{K}, \bar{N}) \cup \Delta(\Sigma_k))$. However, x_n is not allowed to enter the basis until Σ_k is added to RMP, which is possible only if $\bar{c}_k < 0$.

Hence, if the objective function value of the two-stage problem is negative, it means that a variable y_k has negative reduced cost and it enters the basis. Then, $\{x_n | n \in S_N(\Sigma_k)\}$ are allowed to enter the basis. Therefore, with this strategy, we prevent the premature violation in the new dualized constraints $\Delta(\Sigma_k)$ by assigning the values of $\{w_i | i \in \Delta(\Sigma_k)\}$ properly.

We use the above strategy to find the values of the multipliers $\{w_i | i \in \Delta(\Sigma_k)\}$ to solve the PSP correctly. However, in implementing the subgradient optimization algorithm, the new multipliers $\{w_i | i \in \Delta(\Sigma_k)\}$ can also be initialized to zero, which is the convention in the literature. Therefore, the multipliers for the new dualized constraints are initialized either to zero or to the value found by the row-generating PSP. It is stated in [10] that how quickly the subgradient procedure terminates is relatively insensitive to the initial choice of the multipliers. Both approaches converge to the same Lagrangian dual solution but we shall show through our computational experiments that the initialization with the value found by the row-generating subproblem is more beneficial in terms of computation time. The reason is that the values found by the row-generating subproblem act as a penalty to prevent a violation in the new set of linking constraints.

5.2 An Application to The Time-Constrained Routing Problem

We use LR to relax the linking constraints (4.51) using the dual multipliers w_k. Then, the resulting problem is given by

$$
\begin{aligned}
Z(w) = \text{maximize} \quad & \sum_{k \in K} (c_k - w_k) y_k + \sum_{k \in K} w_k \sum_{m \in M} D_{km} x_{km}, && && (5.13)\\
\text{subject to} \quad & \sum_{k \in K} A_{jk} y_k \leq 1, && j \in J, && (5.14)\\
& \sum_{k \in K} B_{km} x_{km} = 1, && m \in M, && (5.15)\\
& y_k \in \{0, 1\}, && k \in K, && (5.16)\\
& x_{km} \in \{0, 1\}, && k \in K, m \in M. && (5.17)
\end{aligned}
$$

where the multipliers w are unrestricted in sign, since the dualized constraints are of equality type. To solve the LP relaxation of this problem by column generation, we replace the set K by $\bar{K}$. Then, the RMP at iteration T of column generation becomes

$$
\begin{aligned}
Z^T(w) = \text{maximize} \quad & \sum_{k \in \bar{K}} (c_k - w_k) y_k + \sum_{k \in \bar{K}} w_k \sum_{m \in M} D_{km} x_{km}, && && (5.18)\\
\text{subject to} \quad & \sum_{k \in \bar{K}} A_{jk} y_k \leq 1, && j \in J, && (5.19)\\
& \sum_{k \in \bar{K}} B_{km} x_{km} = 1, && m \in M, && (5.20)\\
& y_k \in [0, 1], && k \in \bar{K}, && (5.21)\\
& x_{mn} \in [0, 1], && m \in M, n \in N. && (5.22)
\end{aligned}
$$

Note that for a given set of multipliers $\{w_k | k \in \bar{K}\}$, the above problem is separated into two problems. The first one is

$$Z_1^T(w) = \text{maximize} \quad \sum_{k \in \bar{K}} (c_k - w_k) y_k, \tag{5.23}$$

$$\text{subject to} \quad \sum_{k \in \bar{K}} A_{jk} y_k \leq 1, \qquad j \in J, \tag{5.24}$$

$$y_k \in [0, 1], \qquad k \in \bar{K}, \tag{5.25}$$

which is a continuous relaxation of the set-packing problem. The second problem is

$$Z_2^T(w) = \text{maximize} \quad \sum_{k \in \bar{K}} w_k \sum_{m \in M} D_{km} x_{km}, \tag{5.26}$$

$$\text{subject to} \quad \sum_{k \in \bar{K}} B_{km} x_{km} = 1, \qquad m \in M, \tag{5.27}$$

$$x_{mn} \in [0, 1], \qquad m \in M, n \in N. \tag{5.28}$$

The solution of this problem is straightforward, since for each $m \in M$ we can select x_{km} with the maximum objective function coefficient.

The value of the relaxation, $Z^T(w)$, is the summation of the objective function value of these two problems such that $Z^T(w) = Z_1^T(w) + Z_2^T(w)$. Then the Lagrangian dual problem at iteration T of column generation is simply given by

$$Z_{LD}^T = \min_{w_k, k \in \bar{K}} Z^T(w). \tag{5.29}$$

This Lagrangian dual problem is solved and Z_{LD}^T is found by subgradient optimization algorithm. Let $(y^*(w^t), x^*(w^t))$ be the optimal, possibly infeasible, primal solution when (5.23)-(5.25) and (5.26)-(5.28) are solved at iteration t using the multiplier vector $\{w_k^t | k \in \bar{K}\}$. Additionally, let $g_k^t, k \in \bar{K}$ be the subgradient vector associated with the existing relaxed constraints. The entries of this vector are calculated as $g_k^t = (y_k^*(w^t) - \sum_{m \in M} D_{km} x_{km}^*(w^t)),\ k \in \bar{K}$. The values of the multipliers can be updated in a subgradient step as follows:

$$
\begin{aligned}
w_k^{t+1} &= w_k^t + \lambda^t g_k^t, && (5.30)\\
\lambda^t &= \frac{\theta(Z^T(w^t) - LB)}{\sum\limits_{k \in \bar{K}} (g_k^t)^2}, && (5.31)
\end{aligned}
$$

The parameters and the termination criterion for the subgradient optimization are given later when we give the computational results. After finding the solution with subgradient optimization, we solve the row-generating PSP

$$
\zeta_{yx} = \max_{k \in K \setminus \bar{K}} c_k - \sum_{j \in J} A_{jk} u_j - \alpha_k \tag{5.32}
$$

$$
\alpha_k = \text{minimize} \quad w_k \tag{5.33}
$$

$$
\text{subject to} \quad w_k \leq v_m, \quad m \in \{m \in M : B_{km} = 1\}, \tag{5.34}
$$

$$
\text{at least one constraint in (5.34) is tight,} \tag{5.35}
$$

where $\{u_j | j \in J\}$ and $\{v_m | m \in M\}$ are the dual variables corresponding to (5.24) and (5.27), respectively. Note that $\{v_m | m \in M\}$ can simply be obtained by

$$
v_m = \max_{k \in \bar{K} : B_{km} = 1} w_k. \tag{5.36}
$$

Clearly, the model given by (5.32)-(5.35) is quite similar to (4.64)-(4.67). If $\zeta_{yx} < 0$, then y_k, x_{km} for all $m \in M$ such that $B_{km} = 1$ and a new multiplier corresponding to the linking constraint k are added to RMP. The solution of problem (5.33)-(5.35) is $\alpha_k = w_k = \min\limits_{m \in M : B_{km} = 1} v_m$ and the $x_{km}-$variable, for which (5.34) is satisfied with equality, has zero reduced cost. In Chapter 4, this variable is used to augment the optimal basis. However, in CG-LR, the new linking constraint indexed by k is dualized to the objective function with w_k. Hence, allowing x_{km} to enter the basis may violate this constraint. Therefore, we prevent it from entering the basis before y_k enters the RMP.

Next, we conduct computational experiments for TCR problem. We compare the performance of CG-LR against the CRG algorithm given in Chapter 4 on a set of randomly generated instances. In [2], all possible tours are enumerated by checking several feasibility rules. Moreover, some dominance rules are used to speed up the tour enumeration. In this thesis, since we are only concerned with the performances of the algorithms, the tours are generated in a different manner: The number of sites that can be visited are selected as 50 and 150 and the vacation period is taken as 5 days. A single tour can visit at most four sites that are selected randomly. Each site is randomly assigned a rating uniformly distributed between 1 and 100, and the objective function coefficient of a tour is defined as the sum of the ratings of the sites visited in the tour. Moreover, the days in the vacation period in which the tours can take place are also determined randomly. We generate from 10,000 to 100,000 tours.

In contrast to the algorithm in [2], no initial set of tours are selected to form the SRMP. Instead, an artificial variable s_m with an objective function value of $-R$, where R is a large number, is added to the model as follows:

$$\sum_{k \in K} B_{km} x_{km} + s_m = 1, m \in M. \tag{5.37}$$

The CRG algorithm is initialized with only artificial variables and as new columns are added, primal feasibility is achieved and the artificial variables are deleted.

Recall that in Section 5.1 we give a discussion on the use of an upper-bound in the updating phase of the subgradient optimization algorithm. Since TCR is a maximization problem, we need a lower-bound to use in subgradient updating. Three different lower-bounding methods, as mentioned in Section 5.1, are used in our experiments with CG-LR. The first one uses LB, which is the objective function value of some primal feasible solution obtained by a heuristic algorithm. We refer to the results obtained with this lower-bounding as CG-LR1 in Table 5.1. As the second lower-bounding method, we use the objective function value of the previous Lagrangian dual problem, Z_{LD}^{T-1}, as the lower-bound for iteration T of column generation whenever $LB > Z_{LD}^{T-1}$. Otherwise; i.e., when $LB \leq Z_{LD}^{T-1}$, we use LB. The results obtained with this lower-bounding method are given in Table 5.1 as CG-LR2. The last lower-bounding method uses only

Z_{LD}^{T-1} as the lower-bound and its results are listed as CG-LR3 in Table 5.1. To warm start these three CG-LR algorithms, we start with a set of columns that is feasible for (MP). Hence, unlike the CRG algorithm, using artificial variables to initialize the CG-LR algorithms is not necessary. This is done by a brute force procedure, in which the tours are added to the model in their index order until a feasible solution is obtained. Moreover, as mentioned previously, the initial LB can be obtained by a heuristic algorithm. We assume that the value of LB is given by an oracle as $0.95Z$, where Z is the optimal objective function value of (MP). A heuristic algorithm for the TCR could be developed or a rounding algorithm may be called at some points in column generation to find the value of LB.

Moreover, in the application of the subgradient optimization algorithm, we use the following parameters: θ is initialized with value 1.5 and if there is no progress (decrease) in the upper bound for 10 iterations, θ is halved. The subgradient iterations are terminated when 250 iterations are performed or $\theta < 0.05$. Naturally, an increase in these parameters improves the approximation of the multipliers to the optimal dual variables at the expense of increasing computational times.

The results are given in Table 5.1, where "OFV" denotes the objective function value, "Time" denotes the computation time in seconds and "# Col." denotes the number of $y-$variables generated in the column generation algorithm. The optimal objective function value of (MP) is given under the CPLEX column but recall that CRG also finds the optimal value. Objective function values reported for CG-LR algorithms are the approximations of the optimal values, since the stopping criteria of the subgradient optimization are not exact.

Table 5.1: Comparison of algorithms on TCR test instances.

Instances	**CPLEX**		**CRG**		**CG-LR1**			**CG-LR2**			**CG-LR3**		
	OFV	Time	# Col.	Time	OFV	# Col.	Time	OFV	# Col.	Time	OFV	# Col.	Time
10000-50-5-5	1,612.00	3.89	29	1.64	1,612.70	58	40.05	1,612.26	55	40.88	1,708.45	69	90.41
20000-50-5-5	1,605.00	13.45	29	3.05	1,605.36	81	70.08	1,605.35	64	54.72	1,690.51	61	67.45
30000-50-5-5	1,687.00	26.72	29	4.45	1,690.07	90	85.19	1,688.02	75	69.09	1,793.71	58	74.44
40000-50-5-5	1,466.00	49.86	25	5.06	1,467.10	104	154.52	1,466.21	110	123.83	1,519.11	77	86.20
50000-50-5-5	1,584.00	86.46	24	6.03	1,585.56	74	65.22	1,584.72	88	88.52	1,681.32	69	78.16
60000-50-5-5	1,525.00	107.09	24	7.34	1,529.71	120	188.67	1,526.01	92	109.95	1,614.09	74	82.36
70000-50-5-5	1,480.00	130.86	28	9.73	1,481.52	103	148.00	1,481.36	103	142.03	1,574.76	62	96.55
80000-50-5-5	1,568.00	166.27	25	10.00	1,569.20	137	244.41	1,568.24	128	223.23	1,639.75	59	65.95
90000-50-5-5	1,557.00	214.61	27	11.99	1,558.45	89	114.41	1,557.45	131	227.88	1,610.53	63	86.41
100000-50-5-5	1,500.00	280.84	29	14.17	1,501.30	72	77.64	1,501.23	85	104.38	1,598.56	73	98.05
10000-150-5-5	1,805.20	1.22	17	1.00	1,805.63	24	12.95	1,805.61	22	11.61	1,840.30	29	15.95
20000-150-5-5	1,783.00	2.88	13	1.41	1,784.01	21	7.78	1,783.80	21	10.58	1,854.85	43	24.72
30000-150-5-5	1,719.50	5.78	10	1.63	1,720.14	23	11.84	1,720.05	20	10.89	1,840.18	47	51.11
40000-150-5-5	1,862.40	10.09	17	3.48	1,863.41	25	12.91	1,863.27	25	16.95	1,928.93	41	38.38
50000-150-5-5	1,820.67	43.72	20	5.05	1,822.35	27	15.00	1,821.56	25	16.89	1,935.57	61	59.61
60000-150-5-5	1,785.14	24.00	19	5.84	1,786.55	26	15.63	1,786.30	26	17.50	1,847.28	41	41.91
70000-150-5-5	1,880.50	27.69	20	7.08	1,881.61	27	17.63	1,881.23	26	18.02	1,939.57	41	42.27
80000-150-5-5	1,810.60	95.80	23	9.09	1,811.45	32	25.67	1,811.26	29	23.80	1,910.71	69	85.72
90000-150-5-5	1,884.14	58.13	17	7.77	1,885.35	29	26.70	1,884.74	28	24.78	2,024.09	65	79.72
100000-150-5-5	1,842.00	60.23	17	8.48	1,842.98	27	21.89	1,842.99	24	20.28	1,998.87	62	78.78
Average	1,654.82	68.60	22.19	5.94	1,656.17	58.00	65.36	1,655.56	57.43	65.15	1,739.34	56.76	64.71

The results in Table 5.1 indicate that on average, CRG algorithm outperforms other algorithms. As the average solution time figures show CRG is more than ten times faster than CG-LR algorithms. The computation time of CG-LR algorithms are very close to each other. The average of the objective function values obtained by CG-LR1 and CG-LR2 are very close to the average optimal objective function values. Though the indifference is insignificant, CG-LR2 seems to outperform CG-LR1. CG-LR3 is, however, not a strong contender and the average of the objective function values obtained by CG-LR3 is 5% higher than that of CG-LR2. Overall, these findings are in favor of the bounding procedure used in CG-LR2. The gap between the average objective function values of CPLEX and CG-LR2 is just 0.04%. More importantly, the average solution time of CG-LR2 is 5% shorter than that of CPLEX. Note that for the TCR problem, all the algorithms proposed in this thesis use only a very small number of columns.

Note that CG-LR performs poor compared to CRG. The reason is that even though the LR problem is smaller than (MP), after dualizing the linking constraints, it must be solved many times at each subgradient optimization call. On the other hand, solving SRMP by a linear programming solver turned out to be much faster than the subgradient optimization algorithm in our experiments. As discussed in [15] and [48] the use of CG-LR may be justified, if the dual variables found by the simplex algorithm cause many degenerate iterations, and the number of column generation iterations is quite large.

Another question raised above is concerned with the advantages of initializing the multipliers for the new dualized constraints with the values found by the row-generating PSP over initializing them with value zero. The CG-LR algorithms initialize the multipliers for the new dualized constraints with the values found by the row-generating PSP. To this end, we rerun the CG-LR2 algorithm with the new multipliers initialized with value zero. The average values for the objective function and time are 1,656.17 and 109.25, respectively. Even though the average objective function value is close to that of CG-LR2 given in Table 5.1, the average computation time is considerably large. It seems that the initialization of the dual multipliers with the values found by the row-generating PSP is more beneficial.

Chapter 6

CONCLUSIONS AND FUTURE RESEARCH

Column generation is a well-studied concept and has been applied to many problems in different application areas. In some problems that are solved by column generation, the addition of new columns triggers the generation of a set of rows. This simultaneous generation of both columns and rows has been of interest in the literature recently. However, there has been no unified strategy to approach these problems which we refer to as column-dependent-rows problems. In this thesis, we have primarily aimed at characterizing the common aspects of these problems. Then, a unified way to handle these problems and a generic methodology, coined simultaneous column-and-row generation, have been developed. Then, we have applied this solution methodology to time-constrained routing, multi-stage cutting stock and quadratic set covering problems successfully. For the time-constrained routing and multi-stage cutting stock problems, the optimal solution is reached in contrast to the algorithms in the literature. The last problem has been considered in the column generation context for the first time and its analysis has provided us a motivation for making the methodology as generic as possible. Moreover, combination of the proposed column-and-row generation algorithm with Lagrangian relaxation has also been investigated in this thesis. The theoretical results are consolidated by the computational experiments.

The proposed generic algorithms are applied successfully to several example problems in this thesis. As long as the assumptions underlying the CDR problems are satisfied, many other problems can be solved by the proposed solution method. Take for instance a vehicle scheduling problem, where the trip times are exposed to disruptions like delays or trip cancellations. Using a similar approach to that in [61], the resulting problem can be shown to be a column-dependent-rows problem.

Note that the example problems that we consider are actually integer programming problems. We only concentrated on the linear programming relaxation of the problems. After the column-and-row generation algorithm terminates, we can solve an integer program using the set of columns generated. However, this gives only an upper bound on the optimal integral solution. To reach the integral optimal solution, the column-and-row generation algorithm can be embedded in a branch-and-bound scheme.

A follow-up work is the combination of the column-and-row generation algorithm with the Benders decomposition. Using then delayed constraint generation along with column-and-row generation can yield an efficient solution method.

Bibliography

[1] J. M. Van Den Akker, J. A. Hoogeveen, and S. L. Van De Velde. Parallel machine scheduling by column generation. *Operations Research*, 47(6):862–872, 1999.

[2] P. Avella, B. D'Auria, and S. Salerno. A LP-based heuristic for a time-constrained routing problem. *European Journal of Operational Research*, 173:120–124, 2006.

[3] P. Avella, A. Sassano, and I. Vasilev. Computational study of large-scale p-median problems. *Mathematical Programming*, 109:89–114, 2007.

[4] E. Balas and M.C. Carrera. A dynamic subgradient-based branch-and-bound procedure for set covering. *Operations Research*, 44(6):875–890, 1996.

[5] F. Barahona and R. Anbil. The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming*, 87(3):385–399, 2000.

[6] F. Barahona and D. Jensen. Plant location with minimum inventory. *Mathematical Programming*, 83:101–111, 1998.

[7] C. Barnhart, C.A. Hane, and P.H. Vance. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326, 2000.

[8] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[9] M.S. Bazaraa and J.J. Goode. A cutting-plane algorithm for the quadratic set-covering problem. *Operations Research*, 23:150–158, 1975.

[10] J. E. Beasley. Lagrangian relaxation. pages 243–303. John Wiley & Sons, Inc., New York, NY, USA, 1993.

[11] G. Belov and G. Scheithauer. A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. *European Journal of Operational Research*, 171(1):85–106, 2006.

[12] J. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[13] X. Cai, D. C. McKinney, L. S. Lasdon, and Jr. D. W. Watkins. Solving large nonconvex water resources management models using generalized benders decomposition. *Operations Research*, 49:235–245, 2001.

[14] A. Caprara, M. Fischetti, and P. Toth. A heuristic method for the set covering problem. *Operations Research*, 47:730–743, 1999.

[15] D. Cattrysse, M. Salomon, R. Kuik, and L. N. Van Wassenhove. A dual ascent and column generation heuristic for the discrete lotsizing and scheduling problem with setup times. *Management Science*, 39(4):477–486, 1993.

[16] V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics Of Operations Research*, 4(3):233–235, 1979.

[17] A.J. Conejo, E. Castillo, R. Minguez, and G.B. Raquel. *Decomposition Techniques in Mathematical Programming*. Springer, New York, 2006.

[18] J.-F. Cordeau, F. Soumis, and J. Desrosiers. A benders decomposition approach for the locomotive and car assignment problem. *Transportation Science*, 34(2):133–149, 2000.

[19] J.-F. Cordeau, G. Stojkovic, F. Soumis, and J. Desrosiers. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35(4):375–388, 2001.

[20] H. Crowder and M.W. Padberg. Solving large-scale symmetric travelling salesman problems to optimality. *Management Science*, 26(5):495–509, 1980.

[21] G.B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. In T.C. Koopmans, editor, *Activity Analysis of Production and Allocation*, pages 339–347. Wiley & Chapman-Hall, New York-London, 1947.

[22] G.B. Dantzig, D.R. Fulkerson, and S.M. Johnson. Solution of a large-scale travelling-salesman problem. *Operations Research*, 2:393–410, 1954.

[23] G.B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[24] G. Desaulniers, J. Desrosiers, Y. Dumas, S. Marc, B. Rioux, M.M. Solomon, and F. Soumis. Crew pairing at Air France. *European Journal of Operational Research*, 97(2):245–259, 1997.

[25] G. Desaulniers, J. Desrosiers, and M.M. Solomon. *Column Generation*. Springer, New York, 2005.

[26] G. Desaulniers, J. Desrosiers, and S. Spoorendonk. Cutting planes for branch-and-price algorithms. *Les Cahiers du GERAD G-2009-52, HEC Montréal*, Forthcoming in Networks, 2009.

[27] M. Desrochers, J. Desrosiers, and M.M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2):342–354, 1992.

[28] J. Desrosiers and M.E. Lübbecke. Branch-Price-and-Cut Algorithms. In J.J. Cochran, editor, *Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Chichester, 2011.

[29] O. du Merle, D. Villeneuve, Desrosiers J., and Hansen P. Stabilized column generation. *Discrete Mathematics*, 194:229–237, 1999.

[30] S. Elhedhli and J.-L. Goffin. Solution of a large-scale travelling-salesman problem. *Operations Research*, 100(2):267–294, 2004.

[31] S. Elhedhli and J.-L. Goffin. Efficient production-distriburion system design. *Management Science*, 51(7):1151–1164, 2005.

[32] D. Feillet, M. Gendreau, A.L. Medaglia, and J.L. Walteros. A note on branch-and-cut-and-price. *Operations Research Letters*, In Press, 2010.

[33] J.S. Ferreira, M.A. Neves, and P.F. Castro. A two-phase roll cutting problem. *European Journal of Operational Research*, 44(2):185–196, 1990.

[34] L.M. Fisher. The lagrangean relaxation method for solving integer programming problems. *Management Science*, 27(1):1–18, 1981.

[35] A. Frangioni and B. Gendron. 0-1 reformulations of the multicommodity capacitated network design problem. *Discrete Applied Mathematics*, 157:1229–1241, 2009.

[36] M. Gamache, F. Soumis, G. Marquis, and J. Desrosiers. A column generation approach for large-scale aircrew rostering problems. *Operations Research*, 47(2):247–263, 1999.

[37] A. M. Geoffrion and G. W. Graves. Multicommodity distribution system design by benders decomposition. *Management Science*, 20:822–844, 1974.

[38] A.M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10:237–260, 1972.

[39] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9:849–859, 1961.

[40] R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64:275–278, 1958.

[41] M. Guignard. Efficient cuts in lagrangean ‘relax-and-cut’ schemes. *European Journal of Operational Research*, 105(1):216–223, 1998.

[42] R.W. Haessler. A heuristic programming solution to a nonlinear cutting stock problem. *Management Science*, 17(12):793–802, 1971.

[43] M. Held and R.M. Karp. The travelling salesman problem and minimum spanning trees:part i. *Operations Research*, 18:1138–1162, 1970.

[44] M. Held and R.M. Karp. The travelling salesman problem and minimum spanning trees:part ii. *Mathematical Programming*, 1:6–25, 1971.

[45] M. Held, P. Wolfe, and H.P. Crowder. Validation of subgradient optimization. *Mathematical Programming*, 6:62–88, 1974.

[46] K. Hoffman and M.W. Padberg. Solving airline crew scheduling problems by branch and cut. *Management Science*, 39:657–682, 1993.

[47] D. Huisman, R. Jans, M. Peeters, and A.P.M. Wagelmans. Combining column generation and lagrangean relaxation. In G. Desaulniers, J. Desrosiers, and M.M. Solomon, editors, *Column Generation*, chapter 9, pages 247–270. Springer-Verlag, Boston, MA, 2005.

[48] R. Jans and Z. Degraeve. An industrial extension of the discrete lot-sizing and scheduling problem. *IIE Transactions*, 36(1):47 – 58, 2004.

[49] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[50] N. Katayama, M. Chen, and M. Kubo. A capacity scaling heuristic for the multicommodity capacitated network design problem. *Journal of Computational and Applied Mathematics*, 232:90–101, 2009.

[51] L.G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademiia Nauk SSSR*, 244:1093–1096, 1979. (English translation: Soviet Mathematics Doklady, 20(1), 1979).

[52] A. Klose. A lagrangean relax-and-cut approach for the two-stage capacitated facility location problem. *European Journal of Operational Research*, 126(2):408–421, 2000.

[53] N. Kohl and O.B.G. Madsen. An optimization algorithm for the vehicle routing problem with time windows based on lagrangian relaxation. *Operations Research*, 45(3):395–406, 1997.

[54] C. Lemaréchal, J.J. Strodiot, and A. Bihain. On a bundle algorithm for nonsmooth optimization. In O.L. Mangasarian, R.R. Meyer, and S.M. Robinson, editors, *Nonlinear Programming*, chapter 4, pages 331–358. Academic Press, New York, 1981.

[55] M.E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53:1007–1023, 2005.

[56] A. Lucena. Non delayed relax-and-cut algorithms. *Annals of Operations Research*, 140(1):375–410, 2005.

[57] J. Lysgaard, A.N. Letchford, and R.W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100:423–445, 2004.

[58] T.L. Magnanti and R.T. Wong. Accelerated benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29:464–484, 1981.

[59] S. Martello and P. Toth. *Knapsack problems: algorithms and computer implementations.* John Wiley & Sons, Inc., New York, NY, USA, 1990.

[60] İ. Muter, Ş.İ. Birbil, K. Bülbül, and G. Şahin. A note on “a LP-based heuristic for a time-constrained routing problem”. Technical Report SU_FENS_2010/0005, Sabancı University, Istanbul, Turkey, March 2010. http://research.sabanciuniv.edu/14198/.

[61] İ. Muter, Ş.İ. Birbil, K. Bülbül, G. Şahin, D. Taş, D. Tüzün, and H. Yenigün. Solving a robust airline crew pairing problem with column generation. *Computers and Operations Research*, 2010. To appear. `http://dx.doi.org/10.1016/j.cor.2010.11.005`.

[62] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, 1988.

[63] M. Padberg and Rinaldi. G. Facet identification for the symmetric traveling salesman polytope. *Mathematical Programming*, 47:219–257, 1990.

[64] M. Savelsbergh. A branch-and-price algorithm for the generalized assignment problem. *Operations Research*, 45(6):831–841, 1997.

[65] M. Savelsbergh and M. Sol. Drive: Dynamic routing of independent vehicles. *Operations Research*, 46(4):474–490, 1998.

[66] R.R. Saxena and S.R. Arora. A linearization technique for solving the quadratic set covering problem. *Optimization*, 39:33–42, 1997.

[67] P.H. Vance. Branch-and-price algorithms for the one-dimensional cutting stock problem. *Computational Optimization and Applications*, 9:211–228, 1998.

[68] F. Vanderbeck. Implementing mixed integer column generation. In G. Desaulniers, J. Desrosiers, and M.M. Solomon, editors, *Column Generation*, chapter 12, pages 331–358. Springer-Verlag, Boston, MA, 2005.

[69] François Vanderbeck. Computational study of a column generation algorithm for bin packing and cutting stock problems. *Mathematical Programming*, 86:565–594, 1999.

[70] François Vanderbeck. A nested decomposition approach to a three-stage, two-dimensional cutting-stock problem. *Management Science*, 47(6):864–879, 2001.

[71] G. Wang and L. Tang. A row-and-column generation method to a batch machine scheduling problem. In *Proceedings of the Ninth International Symposium*

on Operations Research and Its Applications (ISORA10), page 301308, Chengdu-Jiuzhaigou, China, 2010.

[72] E.J. Zak. Modeling multistage cutting stock problems. *European Journal of Operational Research*, 141:313–327, 2002.

[73] E.J. Zak. Row and column generation technique for a multistage cutting stock problem. *Computers and Operations Research*, 29:1143–1156, 2002.

Curriculum Vitae

Education

- 2007 – 2011, Ph.D., Industrial Engineering, Sabancı University
  Dissertation: Simultaneous Column-and-Row Generation For Solving Large-Scale Linear Programs With Column-Dependent-Rows

- 2002 – 2004, M.S., Operations Management, Celal Bayar University
  Thesis: Design of Experiments for Quality Improvement

- 1995 – 2000, B.S., Industrial Engineering, Dokuz Eylül University
  Graduation Project: An Application of Sales Forecasting

Experience

- Sep 2007–Dec 2011, Teaching Assistant, Sabancı University

- Feb 2010–Jun 2010, Visiting Researcher, University of Bologna, Prof. Andrea Lodi and Prof. Matteo Fischetti
  Research Topic: Chvatal-Gomory Cuts for Integer Programs

- August 2007 – February 2010, Research Assistant, Sabancı University, Project title: Robust Airline Crew Pairing: Models, Solution Techniques and Applications, supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant 106M472

- Feb 2005–Jul 2007, Teaching Assistant, Marmara University

- Feb 2002−Feb 2005, Teaching Assistant, Celal Bayar University
- Jul 2001− Jan 2002, Quality Assurence Systems Consultant, Işık Consulting Ltd.

Papers

- Muter, İ., The Use of Split Plot Designs in Industrial Design of Experiments and the placement of Four - Level Factors into These Designs. *Turkish Statistical Institute, Journal of Statistical Research*,2004,3,1, 21-34.
- Muter, İ., Ş.İ. Birbil, K. Bülbül, G. Şahin, D. Taş, D. Tüzün, H. Yenigün. Solving A Robust Airline Crew Pairing Problem With Column Generation. *Computers and Operations Research*, *To Appear DOI: 10.1016/j.cor.2010.11.005*, 2010.
- Muter, İ., Ş.İ. Birbil, and G. Şahin. Combination of Metaheuristic and Exact Algorithms for Solving Set Covering-Type Optimization Problems. *INFORMS Journal on Computing*, 2010, 22, 4, 603-619.
- Muter, İ., Ş.İ. Birbil, K. Bülbül, G. Şahin. Simultaneous Column-and-Row Generation for Large-Scale Linear Programs with Column-Dependent-Rows, *Minor Revision in Mathematical Programming*, 2010.
- Muter, İ., Ş.İ. Birbil, K. Bülbül, G. Şahin. A Note on "A LP-based Heuristic for a Time-Constrained Routing Problem", *Submitted to European Journal of Operational Research*, 2010.

Proceedings

- Çoban, E., İ. Muter, D. Taş, Ş.İ. Birbil, K. Bülbül, G. Şahin, Y.İ. Topçu, D. Tüzün, H. Yenigün. Column Generation Approaches to a Robust Airline Crew Pairing Model For Managing Extra Flights. *Operations Research Proceedings 2008: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR)*, 2008, 439-444.