

# Chapter 26

## Integrated Column Generation and Lagrangian Relaxation Approach for the Multi-Skill Project Scheduling Problem

Carlos Montoya, Odile Bellenguez-Morineau, Eric Pinson, and David Rivreau

**Abstract** This chapter introduces a procedure to solve the Multi-Skill Project Scheduling Problem. The problem combines both the classical Resource-Constrained Project Scheduling Problem and the multi-purpose machine model. The aim is to find a schedule that minimizes the completion time (makespan) of a project composed of a set of activities. Precedence relations and resources constraints are considered. In this problem, resources are staff members that master several skills. Thus, a given number of workers must be assigned to perform each skill required by an activity. Practical applications include the construction of buildings, as well as production and software development planning. We present an approach that integrates the utilization of Lagrangian relaxation and column generation for obtaining strong makespan lower bounds. Finally, we present the corresponding obtained results.

**Keywords** Column Generation • Lagrangian relaxation • Multi-skilled personnel • Project scheduling • Project staffing • Resource constraints

### 26.1 Introduction

In project scheduling, resource capacity, cost, and resource availabilities play an important role for obtaining a schedule that reaches the goals of a company. Different objectives can be considered in project scheduling such as the maximization of

---

C. Montoya (✉)

Department of Industrial Engineering, Universidad de los Andes, Bogotá D.C, Colombia  
e-mail: [ce.montoya@uniandes.edu.co](mailto:ce.montoya@uniandes.edu.co)

O. Bellenguez-Morineau

IRCCyN, Ecole des Mines de Nantes, Nantes, France  
e-mail: [Odile.morineau@mines-nantes.fr](mailto:Odile.morineau@mines-nantes.fr)

E. Pinson • D. Rivreau

LARIS (EA CNRS 4094), Université Catholique de l'Ouest, Angers, France  
e-mail: [eric.pinson@uco.fr](mailto:eric.pinson@uco.fr); [david.rivreau@uco.fr](mailto:david.rivreau@uco.fr)

© Springer International Publishing Switzerland 2015

C. Schwindt, J. Zimmermann (eds.), *Handbook on Project Management and Scheduling Vol. 1*, International Handbooks on Information Systems,  
DOI 10.1007/978-3-319-05443-8\_26

565

the net present value, the minimization of the resource costs, or the minimization of the project duration (makespan). All of these features are addressed by the resource-constrained project scheduling problems like the RCPSP (Brucker et al. 1999), which are classical scheduling problems that received major attention in the last years.

The RCPSP ( $PS \mid prec \mid C_{max}$ ) deals with a given number of activities that have to be scheduled on a set of resources. It also takes into account precedence relations between activities and limited resource availabilities. The RCPSP considers renewable resources, which can be used whenever they are available (e.g., staff members, machines, and equipment).

The interest in extending the practical applications of the RCPSP encouraged researchers to work on different extensions that capture several variants and features related to specific real life situations (Artigues et al. 2008; Hartmann and Briskorn 2010).

We focus on one particular extension of the RCPSP, which can be classified as a Project Staffing Problem, known as the Multi-Skill Project Scheduling Problem MSPSP ( $PSS \mid prec \mid C_{max}$ , Bellenguez-Morineau and Néron 2005). It is important to mention that the main features of this problem are also described in Chaps. 25 and 28. Furthermore, we can outline that the MSPSP mixes both the RCPSP and the Multi-Purpose Machine model. The aim is to find a schedule that minimizes the makespan of a project, considering that resources are staff members that master several skills. Practical applications include the construction of buildings, production, and software development planning. In addition it is important to notice that this problem is  $\mathcal{NP}$ -hard in the strong sense (Artigues et al. 2008).

In this chapter we propose two Lagrangian relaxation models, which are inspired by a column generation (CG) approach proposed in previous work (Montoya et al. 2013). The proposed approach aims at obtaining strong makespan lower bounds. This chapter is organized as follows: In Sect. 26.2 we present a literature review related to the studied problem and to the methods we use to solve it. In Sect. 26.3 we define the problem and give an overview about column generation and subsequently we introduce two master problem formulations. In Sect. 26.4 we propose two Lagrangian relaxation models. In Sect. 26.5, we introduce the procedures used for initializing the CG procedure and for selecting new columns. In Sect. 26.6 we report the computational results. Finally, in Sect. 26.7 we conclude on our work and discuss possible research avenues.

## 26.2 Literature Review

Before introducing the proposed solution method, we present a literature review related to the studied problem and to the methods we use to solve it.

### **26.2.1 Problem Background**

Despite the fact that the notion of skills plays an important role in the field of personnel assignment (Jiang et al. 2004), it is not often considered in the project scheduling field. Regarding the Multi-Skill Project Scheduling Problem, we refer to the work done by Bellenguez-Morineau and Néron (2005), Bellenguez-Morineau (2008), who developed and implemented different procedures to determine lower and upper bounds for the makespan. More recently, Montoya et al. (2013) proposed a branch-and-price approach for the MSPSP. Other works have been done also on some specific variants for the MSPSP. For example, Cordeau et al. (2010) developed a construction heuristic and an adaptive large neighborhood search heuristic for the Technician and Task Scheduling Problem (TTSP) in a large telecommunications company. The goal in this problem is to assign technicians to tasks with multi-level skill requirements. Here, as it occurs in the MSPSP, the requirements of the tasks (activities) are defined by the presence of a set of resources (technicians) that possess the necessary skills. For solving this last mentioned problem, Firat and Hurkens (2012) developed more recently a solution methodology that uses a flexible matching model as a core engine based on an underlying mixed-integer programming model.

Additionally, there are other interesting solution methodologies in the literature of project scheduling with multi-skilled human resources. For example, Heimerl and Kolisch (2010) proposed a mixed-integer linear program to solve a multi-project problem where the schedule of each project is fixed. They also considered multi-skilled human workforce with heterogeneous and static efficiencies. Li and Womer (2009) developed a hybrid algorithm based on mixed-integer modeling and constraint programming for solving a project scheduling problem with multi-skilled personnel, taking into consideration an individual workload capacity for each worker. Gutjahr et al. (2008) proposed a greedy heuristic and a hybrid solution methodology using priority-based rules, ant colony optimization, and a genetic algorithm to solve the so-called “Project Selection, Scheduling and Staffing with Learning Problem”. More recently, as was introduced in Chap. 25, Correia et al. (2012) presented a mixed-integer linear programming formulation and several sets of additional inequalities for a variant of the resource-constrained project scheduling problem in which resources are flexible i.e., each resource has several skills.

### **26.2.2 Column Generation Overview and Background**

Introduced independently by Dantzig and Wolfe (1960) and Gilmore and Gomory (1961), column generation (CG) consists in solving alternately a (restricted) master problem (RMP) and a sub-problem resulting from the decomposition of the original problem. The main idea underlying this approach is to select, at each iteration of

an iterative process and by means of an oracle related to the column generation sub-problem and using duality information originating from the solving of the current RMP, a candidate variable whose reduced cost is susceptible to improve the objective function associated with the master problem. The related column is added to the current RMP, and we iterate until no more candidate can be found. Notice that such a decomposition is possible due to the exploitation of some specific structure of the problem formulation whose pricing sub-problem leads to an “easier” optimization problem such as shortest path or knapsack problems.

So far column generation (CG) had been used to solve specifically the Multi-Skill Project Scheduling Problem by Montoya et al. (2013). Additionally CG has been used in combination with other optimization techniques for solving project scheduling problems. Particularly, Brucker et al. (1999) implemented a destructive approach for finding tight lower bounds for the RCPSP by using both constraint propagation techniques and CG. Afterwards, authors extended their work for solving the Multi-Mode RCPSP (MRCPSP) (Brucker and Knust 2000). Additionally, Van den Akker et al. (2007) presented a destructive lower bound based on column generation for certain extensions of the RCPSP. In this approach, authors used a simulated annealing algorithm for finding a schedule for each resource, which was enforced by a time-indexed integer programming formulation.

On the other hand, CG has been widely used on the Vehicle Routing Problem (VRP) and several related extensions (Lübbecke and Desrosiers 2005). Some VRP problems consider similar features to those of the MSPSP. For example, Dohn et al. (2009) deal with an assignment problem where a set of teams must be assigned to a set of tasks, restricted by time-windows. As it occurs in the MSPSP, assigned resources must start and finish a given activity simultaneously. Authors developed a branch-and-price procedure and enforce the fulfillment of such a constraint with a branching scheme that limits the starting time of a given activity. Moreover, for a particular extension of the VRP, Ioachim et al. (1999) modeled the synchronization constraint directly in the master problem with the consequence that a large number of columns with a small variation in the starting times (departure times) of the tasks (flights) are generated. To handle such a drawback, they introduced a tolerance in the side constraints to allow asynchronous departure times.

Additionally, column generation has also been used to solve Staff Scheduling Problems (Bard and Purnomo 2005; Beliën and Demeulemeester 2007; Jaumard et al. 1998; Mason and Smith 1998; Mehrotra et al. 2000). This type of problems, as it occurs with the MSPSP, involves the assignment of staff members to perform a set of activities, but they normally intend to minimize a total assignment cost, considering also a predefined time horizon.

Finally, CG has also been used to solve shop scheduling problems (Chen and Powell 1999; Gélinas and Soumis 2005; Van den Akker et al. 2000, 2002). These problems are related to single machine, flexible and job shop scheduling problems, sharing also some common features with the MSPSP. For example, Gélinas and Soumis (2005) deal with precedence constraints for solving the job shop scheduling problem. They handle these constraints in the master problem and modeled the

sub-problem as a single machine problem with time constraints. On another side, Van den Akker et al. (2000) used CG based on a time-indexed formulation for solving single machine scheduling problems. They used Dantzig-Wolfe decomposition techniques (Dantzig and Wolfe 1960) to deal with the difficulties related to the size of a time-indexed formulation, given its capacity to obtain strong lower bounds. This approach supports the one considered in this chapter, since our work is also based on a time-indexed perspective.

### ***26.2.3 Combining Lagrangian Relaxation and Column Generation Background***

Typically, column generation is used to solve the LP-relaxation of the master problem, but it can also be combined with Lagrangian relaxation as we will discuss in this chapter. According to Wolsey (1998), it is possible to solve the Lagrangian dual either by means of the subgradient method or by solving the linear relaxation of the extensive formulation by using a CG approach. Thereafter, the optimal lower bound of the restricted linear master problem (RMP) and the best Lagrangian dual will have the same value. Both solution methods for the Lagrangian dual have advantages and disadvantages, hence some authors have proposed procedures that try to combine the advantages of both approaches (Barahona and Jensen 1998; Huisman et al. 2005).

As we will show later on, each Lagrangian multiplier vector is linked with the dual variables related to the relaxed constraint. Consequently, this implies that the dual values obtained by solving the RMP can be estimated by the Lagrangian multipliers used in the related Lagrangian dual problem. Thus, instead of solving the RMP with the simplex method by using a solver, we can use a subgradient procedure (Held and Karp 1971) for solving the Lagrangian dual approximately. The associated Lagrangian multipliers can be used for estimating the values of the dual variables related to the constraints of the RMP and for pricing out new columns.

Overall, there are different reasons for using this last mentioned approach. The subgradient method is fast, easy to implement, and does not require a commercial solver. When solving the RMP with a simplex method, we obtain a basic dual solution that corresponds to a vertex of the optimal face of the dual polyhedron. Given that a new column of the RMP may cut that vertex, a dual solution interior (in the center) of the dual face allows stronger dual cuts (i.e., better primal columns). Bixby et al. (1992) and Barnhart et al. (1998) obtained from their research that this may improve the convergence of a column generation algorithm and reduce degeneracy. Jans and Degraeve (2004) and Huisman et al. (2005) provide computational results that indicate that Lagrangian multipliers are beneficial. Finally, it is also shown that during the subgradient phase, possible feasible solutions are generated.

## 26.3 Problem Description and Column Generation

As was already mentioned, the Multi-Skill Project Scheduling Problem MSPSP is a known project scheduling problem, which is mainly composed by three components: activities, resources, and skills. It considers a set  $V$  of  $n$  activities. Within this set, we also define two dummy activities 0 and  $n + 1$  which represent respectively the beginning and completion of the project. Let  $Succ(i)$  denote the set of immediate successors of an activity  $i$  whose processing time is denoted by  $p_i$ . Additionally, a set  $\mathcal{R}$  of  $K$  workers and a set  $\mathcal{L}$  of  $L$  skills are defined for performing these activities. We denote by  $r_{il}$  the number of workers with skill  $l$  required by activity  $i$ . We define  $a_{kl} = 1$  if worker  $k$  masters skill  $l$ , 0 otherwise. Finally,  $T$  denotes an upper bound for the total duration of the project or makespan.

After understanding the main features of the MSPSP, we introduce the two master problem formulations related to the two Lagrangian relaxation models exploited in this chapter.

### 26.3.1 Column Generation Master Problem Formulations

The basic idea underlying the considered CG approach relies on a time-indexed reformulation of the problem. In this mathematical formalization, a column  $\omega$  (i.e., activity work pattern) represents certain processing attributes (a starting time and set of assigned workers) of an activity  $i$ . Hence, a solution to the MSPSP consists in identifying a single column (a starting time and a set of assigned workers) for each activity of the project. An activity work pattern  $\omega$  is represented by three parameters: (i)  $\epsilon_i^\omega$ , which takes the value of 1 if activity  $i$  is processed in activity work pattern  $\omega$ , 0 otherwise; (ii)  $\gamma_i^\omega$ , which takes the value of  $t$  if activity  $i$  starts at time  $t$  in activity pattern  $\omega$ , 0 otherwise; (iii)  $\zeta_{kt}^\omega$ , which takes the value of 1 if worker  $k$  is assigned on activity work pattern  $\omega$  at time  $t$ , 0 otherwise. We assume that the workers assigned to  $\omega$  satisfy the skills requirement of the related activity. Additionally, we denote the set of all feasible activity work patterns by  $\Omega$ .

The decision variables governing the target model are defined by: (i)  $x_\omega$ , which takes the value of 1 if activity work pattern  $\omega$  is selected, 0 otherwise; (ii)  $S_i$ , which represents the starting time of an activity  $i$ .

Based on the previous description we present two master problem formulations ( $MP_1$  and  $MP_2$ ), that lead to the same sub-problem (SP). Moreover, we can state that solving the SP aims to exhibit a feasible selection of workers/skills for processing activity  $i$  at time  $t$ .

In the next subsections we introduce  $MP_1$  and  $MP_2$ , and later on, we explain the sub-problem and the solution method applied to solve it.

### 26.3.1.1 First Master Problem ( $MP_1$ )

The first master problem  $MP_1$  can be stated as follows:

$$Z[MP_1]: \text{Min. } S_{n+1} \quad (26.1)$$

$$\text{s. t. } \sum_{\omega \in \Omega} (x_\omega \cdot \epsilon_i^\omega) = 1 \quad (i \in V) \quad (26.2)$$

$$\sum_{\omega \in \Omega} (x_\omega \cdot \Upsilon_i^\omega) = S_i \quad (i \in V) \quad (26.3)$$

$$\sum_{\omega \in \Omega} (x_\omega \cdot \zeta_{kt}^\omega) \leq 1 \quad (k \in \mathcal{R}; t = 0, \dots, T) \quad (26.4)$$

$$S_i + p_i \leq S_j \quad (i \in V; j \in \text{Succ}(i)) \quad (26.5)$$

$$ES_i \leq S_i \leq LS_i \quad (i \in V) \quad (26.6)$$

$$x_\omega \in \{0, 1\} \quad (\omega \in \Omega) \quad (26.7)$$

The objective is to minimize the makespan (26.1). Constraint set (26.2) ensures that only a unique activity work pattern can be assigned to any task  $i$ . Constraints (26.3) recover the associated starting times, while constraint set (26.4) ensures that any operator can carry out at most one activity at a given time. Constraints (26.5) state the precedence relations connecting the activities in  $V$ , and constraint set (26.6) ensures that the starting time of each activity must be within a predefined time-window. For this purpose,  $ES_i$  (resp.  $LS_i$ ) denotes in this formulation a lower bound (resp. upper) for the starting date associated with activity  $i$ . Such a time-window is for instance simply induced by the precedence graph using recursively Bellman's conditions and a given upper bound ( $T$ ) for the makespan.

Thereafter, for applying CG, integrality constraints (26.7) are relaxed. The (restricted) master problem ( $RMP_1$ ) is then obtained by considering a partial pool of activity work patterns  $\bar{\Omega}$  ( $\bar{\Omega} \subseteq \Omega$ ). Assuming that an optimal solution of the  $RMP_1$  has been computed with a standard LP solver, let us denote the simplex multipliers associated with constraints (26.2), (26.3), and (26.4) respectively by  $(\rho_i, i \in V)$ ,  $(\tau_i, i \in V)$ , and  $(\theta_{kt}, k \in \mathcal{R}, t = 0, \dots, T)$ . The reduced cost associated with an activity work pattern  $\omega$  related to the processing of activity  $i$  at time  $t$  can be stated as follows:

$$rc_{it} = 0 - \rho_i - (\tau_i \cdot t) - \sum_{k \in \mathcal{R}} \sum_{t'=t}^{t'+p_i-1} (\theta_{kt'} \cdot \zeta_{kt'}^\omega) = rc_{it}^1 + rc_{it}^2 \quad (26.8)$$

with:

$$rc_{it}^1 = -\rho_i - (\tau_i \cdot t) \quad (26.9)$$

$$rc_{it}^2 = - \sum_{k \in \mathcal{R}} \sum_{t'=t}^{t'+p_i-1} (\theta_{kt'} \cdot \zeta_{kt'}^\omega) \tag{26.10}$$

Hence, if  $rc_{it} < 0$ , then the corresponding column (activity work pattern) can be added to the current pool of columns, iteratively, until no more profitable columns can be found (Gilmore and Gomory 1961).

**26.3.1.2 Second Master Problem (MP<sub>2</sub>)**

This new master problem formulation has a similar structure to the one previously explained. This new mathematical model relies on an alternative way for integrating the precedence relations constraints. Hence, let us consider a particular precedence relation between two activities  $i$  and  $j$  ( $j \in Succ(i)$ ) which must be processed in the time slot  $W_{ij} = ES_j, \dots, LS_j + p_j$ . Mapping this time slot, we can extend the notion of an activity work pattern  $\omega$  by defining a new parameter  $v_{ijt'}^\omega$  in the following way:

If  $\omega$  corresponds to the execution of  $i$  at a starting time  $t$ :

- $v_{ijt'}^\omega = 1$  ( $t' = ES_i, \dots, t + p_i - 1$ );
- $v_{ijt'}^\omega = 0$  ( $t' = t + p_i, \dots, LS_j + p_j$ ).

If  $\omega$  corresponds to the execution of  $j$  at a starting time  $t$ :

- $v_{ijt'}^\omega = 1$  ( $t' = ES_i, \dots, t - 1$ );
- $v_{ijt'}^\omega = 0$  ( $t' = t, \dots, LS_j + p_j$ ).

If  $\omega$  does not correspond to the execution of neither  $i$  nor  $j$ :

- $v_{ijt'}^\omega = 0$  ( $t' = ES_i, \dots, LS_j + p_j$ ).

Clearly, two activity work patterns  $\omega_1$  and  $\omega_2$  should be consistent for the precedence constraint of  $i$  and  $j$ , if:

$$(v_{ijt'}^{\omega_1} \cdot x_{\omega_1}) + (v_{ijt'}^{\omega_2} \cdot x_{\omega_2}) \leq 1 \quad (t' \in W_{ij}) \tag{26.11}$$

Additionally, we also consider a coefficient  $C_{max}^\omega$ , related to each activity work pattern  $\omega$ . Hence, given that  $S(\omega)$  represents the starting time linked to activity work pattern  $\omega$ , we define  $C_{max}^\omega$  as follows:

- $C_{max}^\omega = S(\omega)$  if activity work pattern  $\omega$  is related to the execution of the dummy activity  $n + 1$ , 0 otherwise.

Notice that the proposed mathematical formulations are adapted for most of regular as well as nonregular optimization criteria.

Let us also recall that  $\Omega$  represents the set of all feasible activity work patterns. The only decision variable governing the target model is  $x_\omega$ , which was already



introduced for  $MP_1$ . Therefore, the associated mathematical formulation can then be stated as follows:

$$Z[MP_2]: \text{Min. } \sum_{\omega \in \Omega} (C_{max}^\omega \cdot x_\omega) \tag{26.12}$$

$$\text{s. t. } \sum_{\omega \in \Omega} (x_\omega \cdot \epsilon_i^\omega) = 1 \quad (i \in V) \tag{26.13}$$

$$\sum_{\omega \in \Omega} (x_\omega \cdot \zeta_{kt}^\omega) \leq 1 \quad (k \in \mathcal{R}; t = 0, \dots, T) \tag{26.14}$$

$$\sum_{\omega \in \Omega} (x_\omega \cdot v_{ijt}^\omega) \leq 1 \quad (i \in V; j \in Succ(i); t \in W_{ij}) \tag{26.15}$$

$$x_\omega \in \{0, 1\} \quad (\omega \in \Omega) \tag{26.16}$$

Notice that this formulation has a structure (set packing problem) with a pure 0-1 coefficients constraint matrix. More precisely, we have that constraint set (26.13) ensures that only a unique activity work pattern can be assigned to any task  $i$ . Constraint set (26.14) ensures that any operator can carry out at most one activity at a given time. Constraint (26.15) states the precedence relations connecting the activities in  $V$  at given time-point  $t$ .

The (restricted) master problem ( $RMP_2$ ) can simply be obtained by relaxing the binarity constraints relating to decision variables  $x_\omega$  and by considering a partial pool of activity work patterns  $\bar{\Omega}$  ( $\bar{\Omega} \subseteq \Omega$ ). Thus, after solving the  $RMP_2$  the corresponding simplex multipliers (dual variables) associated to constraints (26.13), (26.14), (26.15) are denoted respectively by  $(\rho_i, i \in V)$ ,  $(\theta_{kt}, k \in \mathcal{R}, t = 0, \dots, T)$ , and  $(\eta_{ijt}, i \in V, j \in Succ(i), t \in W_{ij})$ . Subsequently, the reduced cost associated with an activity work pattern  $\omega$  related to the processing of activity  $i$  at time  $t$  can be stated as follows:

$$rc_{it} = 0 - \rho_i - \sum_{i \in V} \sum_{j \in Succ(i)} \sum_{t' \in W_{ij}} (v_{ijt'}^\omega \cdot \eta_{ijt'}) - \sum_{k \in \mathcal{R}} \sum_{t'=t}^{t'+p_i-1} (\theta_{kt'} \cdot \zeta_{kt'}^\omega) = rc_{it}^1 + rc_{it}^2 \tag{26.17}$$

with:

$$rc_{it}^1 = -\rho_i - \sum_{i \in V} \sum_{j \in Succ(i)} \sum_{t' \in W_{ij}} (v_{ijt'}^\omega \cdot \eta_{ijt'}) \tag{26.18}$$

$$rc_{it}^2 = - \sum_{k \in \mathcal{R}} \sum_{t'=t}^{t'+p_i-1} (\theta_{kt'} \cdot \zeta_{kt'}^\omega) \tag{26.19}$$

As was already mentioned a column is added to the current pool of columns, iteratively, until no more profitable columns can be found (Gilmore and Gomory 1961).

Now, before introducing the combined Lagrangian relaxation and column generation approaches proposed in the present chapter, we give an insight to the sub-problem (SP) and to the applied solution method.

### 26.3.1.3 Column Generation Sub-Problem (SP)

Assuming that an optimal solution of  $RMP_1$  or  $RMP_2$  has been computed, let us define the quantity  $c_{kt}$ , which simply corresponds to the total cost incurred by worker  $k$  when assigned to an activity  $i$  in the time slot  $t, \dots, t + p_i - 1$ .

$$c_{kt} = - \sum_{t'=t}^{t+p_i-1} \theta_{kt'} \tag{26.20}$$

Subsequently, we consider the decision variables  $y_k = 1$ , if worker  $k$  is assigned to perform activity  $i$ , 0, otherwise, and  $z_{kl} = 1$ , if worker  $k$  uses skill  $l$  to perform activity  $i$ , 0 otherwise. Finding an optimal feasible selection of workers/skills for processing activity  $i$  at time  $t$  that minimizes the reduced cost  $rc_{it}$  leads to the following sub-problem  $SP(i, t)$ :

$$Z[SP(i, t)]: \text{Min. } rc_{it}^2 = \sum_{k \in \mathcal{R}} (c_{kt} \cdot y_k) \tag{26.21}$$

$$\text{s. t. } \sum_{k \in \mathcal{R}} z_{kl} = r_{il} \quad (l \in \mathcal{L}) \tag{26.22}$$

$$y_k = \sum_{l \in \mathcal{L}} z_{kl} \quad (k \in \mathcal{R}) \tag{26.23}$$

$$y_k \in \{0, 1\}, z_{kl} \in \{0, 1\} \quad (k \in \mathcal{R}; l \in \mathcal{L}) \tag{26.24}$$

In this formulation, the objective is to minimize the total assignment cost to perform activity  $i$  at a time  $t$ . Constraint set (26.22) guarantees its requirements fulfillment. Constraint set (26.23) ensures that an assigned worker uses only one skill. Finally, constraint set (26.24) defines the decision variables as binary. Moreover, we can state that solving the SP aims to exhibit a feasible selection of workers/skills for processing activity  $i$  at time  $t$ .

Once  $rc_{it}^2 = -Z[SP(i, t)]$  is computed, we get the targeted reduced cost defined by (26.8) and (26.17) related to the solution of the  $RMP_1$  and  $RMP_2$  respectively. If  $rc_{it} < 0$ , then the corresponding column is candidate to enter the basis since it leads to a decrease in the objective function value for the corresponding restricted master problem ( $RMP_1$  or  $RMP_2$ ). Consequently, this activity work pattern can be added to

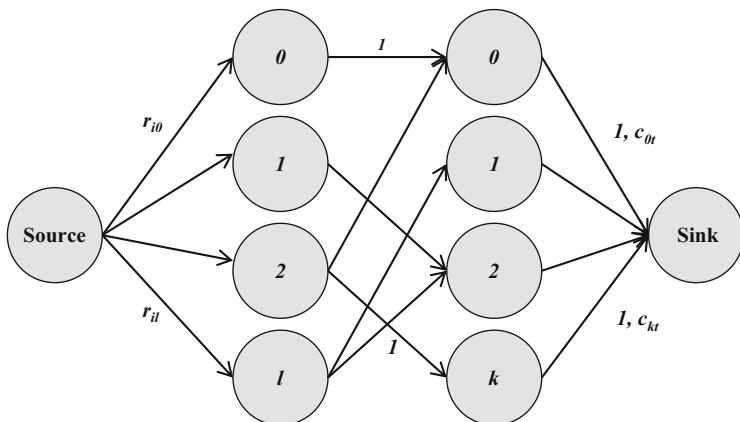


Fig. 26.1 Graph  $G(i, t)$  skills assignment for activity  $i$

the current pool of columns according to the parameters defined for each of the two proposed master problem formulations in Sects. 26.3.1.1 and 26.3.1.2.

Of course, an enumeration on each activity for each of its potential starting date ( $ES_i \leq t_i \leq LS_i$ ) is necessary for exhibiting an activity work pattern with global minimal reduced cost. We refer to the next section for more details related to the global solution method.

### 26.3.1.4 Column Generation Sub-Problem Solution (SP)

Clearly, solving  $SP(i, t)$  is equivalent to find an optimal solution to a min-cost max-flow problem on a particular network  $G(i, t)$ , whose structure is depicted in Fig. 26.1. This graph simply formalizes the assignment of the skills required for performing activity  $i$  to the workers mastering at least one of these skills, according to the constraint stating that any worker can use only one skill when performing a given activity. The values on each arc correspond respectively to its flow upper bound and the related unit cost. Notice that each arc  $(l, sink)$  with a positive flow corresponds to a selected worker for the processing of activity  $i$  ( $y_k = 1$ ). This classical flow problem can be efficiently solved using the algorithm proposed by Busacker and Gowen (1961) in  $\mathcal{O}((K + L)^3)$  time complexity.

## 26.4 Combining Lagrangian Relaxation and Column Generation

One of the main proposal of this chapter relies on combining CG with Lagrangian relaxation, leading to a faster way of solving the (restricted) master problem rather than using an LP solver (Huisman et al. 2005). In the sequel, we present the

two Lagrangian relaxation models related to the solution of  $RMP_1$  and  $RMP_2$ , respectively.

### 26.4.1 $RMP_1$ Based Model for Combining Lagrangian Relaxation and Column Generation

Before introducing the Lagrangian relaxation model related to  $RMP_1$ , we include the next additional surrogate constraint to the (restricted) master problem described previously:

$$\sum_{t=0, \dots, T} \sum_{k \in \mathcal{R}} \sum_{\omega \in \tilde{\Omega}} (x_\omega \cdot \zeta_{kt}^\omega) = \sum_{i \in V} \sum_{l \in \mathcal{L}} (p_i \cdot r_{il}) \tag{26.25}$$

Such a constraint establishes that the accumulated time per resource unit assigned (left term) must be equal to the total amount of time per resource unit required during the whole project duration (right term). Thereafter, let us associate with constraints (26.2), (26.3), and (26.4) the respective Lagrangian multipliers  $(\rho_i, i \in V)$ ,  $(\tau_i, i \in V)$  and  $(\theta_{kt}, k \in \mathcal{R}, t = 0, \dots, T)$ . The corresponding Lagrangian function can be written as follows:

$$\begin{aligned} \Gamma(x, t, \rho, \tau, \theta) = & S_{n+1} + \sum_{i \in V} \rho_i \cdot (1 - \sum_{\omega \in \Omega} (x_\omega \cdot \epsilon_i^\omega)) \\ & + \sum_{i \in V} \tau_i \cdot (S_i - \sum_{\omega \in \Omega} (x_\omega \cdot \Upsilon_i^\omega)) + \sum_{k \in \mathcal{R}} \sum_{t=0, \dots, T} \theta_{kt} \cdot (1 - \sum_{\omega \in \Omega} (x_\omega \cdot \zeta_{kt}^\omega)) \end{aligned} \tag{26.26}$$

$$\begin{aligned} \Gamma(x, t, \rho, \tau, \theta) = & S_{n+1} + \sum_{i \in V} (\tau_i \cdot S_i) + \sum_{\omega \in \Omega} ((-\epsilon_i^\omega \cdot \rho_i) - (\Upsilon_i^\omega \cdot \tau_i)) \\ & - \sum_{k \in \mathcal{R}} \sum_{t=0, \dots, T} (\theta_{kt} \cdot \zeta_{kt}^\omega) \cdot x_\omega + \sum_{i \in V} \rho_i + \sum_{k \in \mathcal{R}} \sum_{t=0, \dots, T} \theta_{kt} \end{aligned} \tag{26.27}$$

For a given distribution  $(\rho, \tau, \theta)$  of Lagrangian multipliers, the associated dual function  $L(\rho, \tau, \theta)$  can be computed solving the following independent Lagrangian sub-problems:

$$Z[LSP_1(\tau)]: \text{Min. } S_{n+1} + \sum_{i \in V} (\tau_i \cdot S_i) \tag{26.28}$$

$$\text{s. t. } S_i + p_i \leq S_j \quad (i \in V; j \in \text{Succ}(i)) \tag{26.29}$$

$$ES_i \leq S_i \leq LS_i \quad (i \in V) \tag{26.30}$$

and

$$Z[LSP_2(\rho, \tau, \theta, x)]: \text{Min. } \sum_{\omega \in \Omega} (C'_{max}{}^\omega \cdot x_\omega) \quad (26.31)$$

$$\text{s. t. } \sum_{\omega \in \Omega} (x_\omega \cdot wl_\omega) = \sum_{i \in V} \sum_{l \in \mathcal{L}} (p_i \cdot r_{il}) \quad (26.32)$$

$$x_\omega \in \{0, 1\} \quad (\omega \in \Omega) \quad (26.33)$$

where

$$C'_{max}{}^\omega = -\rho_{i(\omega)} - (S(\omega) \cdot \tau_i) - \sum_{k \in \mathcal{R}} \sum_{t=0, \dots, T} (\theta_{kt} \cdot \zeta_{kt}^\omega) \quad (26.34)$$

$$wl_\omega = \sum_{l \in \mathcal{L}} (p_{i(\omega)} \cdot r_{il}) \quad (26.35)$$

Hence, obviously we have:

$$\begin{aligned} Z[L(\rho, \tau, \theta)] &= Z[LSP_1(\rho, \tau, \theta, t)] \\ &+ Z[LSP_2(\rho, \tau, \theta, x)] + \sum_{i \in V} \rho_i + \sum_{k \in \mathcal{R}} \sum_{t=0, \dots, T} \theta_{kt} \end{aligned} \quad (26.36)$$

Subsequently, we use a commercial solver for the solution of the first Lagrangian sub-problem ( $LSP_1(\tau)$ ). In addition, it can be noticed that  $LSP_2(\rho, \tau, \theta)$ , is a classical 0-1 knapsack problem, which can be quite time consuming when the pool of patterns  $\bar{\Omega}$  increases. This problem can be solved with a pseudo-polynomial time complexity of  $\mathcal{O}(qB)$ , where:

$$q = |\bar{\Omega}| \quad (26.37)$$

$$B = \sum_{i \in V} \sum_{l \in \mathcal{L}} (p_i \cdot r_{il}) \quad (26.38)$$

Nevertheless, we can focus on the linear relaxation of  $LSP_2$  according to the activity work pattern generation process ( $x_\omega \geq 0, \omega \in \bar{\Omega}$ ). First, let us sort the activity work pattern in  $\bar{\Omega}$  in such a way that:

$$C'_{max}{}^{\omega_1} / wl_{\omega_1} \leq C'_{max}{}^{\omega_2} / wl_{\omega_2} \leq \dots \leq C'_{max}{}^{\omega_q} / wl_{\omega_q} \quad (26.39)$$

Now, after sorting activity work patterns according to (26.39), let  $s$  be the maximal index such that:

$$\sum_{j=0}^s wl_{\omega_j} \leq \sum_{i \in V} \sum_{l \in \mathcal{L}} (p_i \cdot r_{il}) \quad (26.40)$$

An optimal solution to  $LSP_2$  is given by:

$$\bar{x}_{\omega_j} = 1 \quad (j = 0, \dots, s) \tag{26.41}$$

$$\bar{x}_{\omega_{s+1}} = \frac{(\sum_{i \in V} \sum_{l \in \mathcal{L}} (p_i \cdot r_{il})) - w l_{\omega_j}}{w l_{\omega_{s+1}}} \tag{26.42}$$

$$\bar{x}_{\omega_j} = 0 \quad (j = s + 2, \dots, q) \tag{26.43}$$

Considering that  $Z[L(\rho, \tau, \theta)]$  defines a lower bound on the  $RMP_1$ , we can obtain the best possible lower bound by solving the related Lagrangian dual problem ( $LDRMP_1$ ):

$$Z[LDRMP_1] = \text{Max}_{\rho, \tau, \theta} L(\rho, \tau, \theta) \tag{26.44}$$

In the context of combinatorial optimization one efficient way to solve the Lagrangian dual problem is to use a subgradient procedure introduced by Held and Karp (1971), which iteratively updates the Lagrangian multipliers. Nevertheless other methods like volume (Barahona and Anbil 2000), bundle (Fábián 2000), and analytic center cutting plane methods (Goffin and Vial 2002) among others (Bertsekas 1999) can be used for solving the Lagrangian dual problem. We focus particularly on the description of the subgradient method since it is the most diffused one. In addition, besides the fact that it was the first one used in the context of combinatorial optimization, it has, at least, two main advantages: it is easy to code and has minimal memory requirements.

Thereafter, we use the subgradient procedure for estimating the values of the Lagrangian multipliers  $(\rho, \tau, \theta)$ . Hence, after solving  $LSP_1(\tau)$  and  $LSP_2(\rho, \tau, \theta)$  we obtain the starting times vector  $S$  and the column assignment vector  $\bar{x}$  from the solution of the respective sub-problem. Consequently, we can define a subgradient for  $L(\rho, \tau, \theta)$  as follows:

$$\psi_i^1 = 1 - \sum_{\omega \in \Omega} (x_\omega \cdot \epsilon_i^\omega) \quad (i \in V) \tag{26.45}$$

$$\psi_i^2 = S_i - \sum_{\omega \in \Omega} (x_\omega \cdot \gamma_i^\omega) \quad (i \in V) \tag{26.46}$$

$$\varphi_{kt} = (1 - \sum_{\omega \in \Omega} (x_\omega \cdot \zeta_{kt}^\omega)) \quad (k \in \mathcal{R}; t = 0, \dots, T) \tag{26.47}$$

Now, given an upper bound  $T$  for the makespan and a step size  $sp$ , we can update the current Lagrangian multipliers by:

$$\rho_i = \rho_i + (sp \cdot \psi_i^1) \quad (i \in V) \tag{26.48}$$

$$\tau_i = \tau_i + (sp \cdot \psi_i^2) \quad (i \in V) \tag{26.49}$$

$$\theta_{kt} = \min\{0, \theta_{kt} + (sp \cdot \varphi_{kt})\} \quad (k \in \mathcal{R}; t = 0, \dots, T) \tag{26.50}$$

The step size  $sp$  is defined as follows:

$$sp = vp \cdot \frac{(T - L(\rho, \tau, \theta))}{Norm} \quad (26.51)$$

where  $Norm$  is given by:

$$Norm = \sum_{i \in V} (\psi_i^1 + \psi_i^2)^2 + \sum_{k \in \mathcal{R}} \sum_{t=0, \dots, T} \varphi_{kt}^2 \quad (26.52)$$

Equation (26.51) represents a known step length rule, which was empirically justified by Held et al. (1974). Moreover, this rule is less expensive in terms of CPU times in comparison to other step length rules with proven convergence (Polyak 1967). Notice that in this equation we consider a parameter  $vp$ , which is initialized with a value equal to 2 ( $vp = 2$ ), as was proposed by Held and Karp (1971). Additionally it is important to set a limit on the number of iterations, which defines also the accuracy of the solution.

Overall, the general idea is to update the Lagrangian multipliers during a limited number of iterations. Thereafter, we use the last updated multipliers as an estimation of the dual multipliers for pricing out new columns. At the end of each iteration we update the parameter  $vp$  with a systematic geometric revision:  $vp = gp \cdot vp$ . Normally the second parameter  $gp$  ranges between 0.87 and 0.9995, depending on the targeting problem.

Finally, when there are no more columns with a negative reduced cost by using the Lagrangian multipliers, we perform a fixed number of iterations solving the  $RMP_1$  with the simplex method by using the solver for obtaining the values of the dual variables for pricing out new columns. Hence, if after the fixed number of iterations there are still columns with negative reduced costs, we go back to the Lagrangian procedure, otherwise, we stop.

## 26.4.2 $RMP_2$ Based Model for Combining Lagrangian Relaxation and Column Generation

The second Lagrangian model proposed is based on the second (restricted) master problem formulation ( $RMP_2$ ) proposed in Sect. 26.3.1.2. Now, let us associate with constraints (26.13), (26.15), and (26.14) the respective Lagrangian multipliers ( $\rho_i, i \in V$ ), ( $\eta_{ijt}, i \in V, j \in Succ(i), t \in W_{ij}$ ), and ( $\theta_{kt}, k \in \mathcal{R}, t = 0, \dots, T$ ). The corresponding Lagrangian function can be written as follows:

$$\Gamma(x, \rho, \eta, \theta) = \sum_{\omega \in \Omega} (C_{max}^\omega \cdot x_\omega) + \sum_{i \in V} \rho_i \cdot (1 - \sum_{\omega \in \Omega} (x_\omega \cdot \epsilon_i^\omega))$$

$$\begin{aligned}
& + \sum_{i \in V} \sum_{j \in \text{Succ}(i)} \sum_{t \in W_{ij}} \eta_{ijt} \cdot (1 - \sum_{\omega \in \Omega} (x_{\omega} \cdot v_{ijt}^{\omega})) \\
& + \sum_{k \in \mathcal{R}} \sum_{t=0, \dots, T} \theta_{kt} \cdot (1 - \sum_{\omega \in \Omega} (x_{\omega} \cdot \zeta_{kt}^{\omega})) \quad (26.53)
\end{aligned}$$

$$\begin{aligned}
\Gamma(x, \rho, \eta, \theta) = & \sum_{\omega \in \Omega} (C_{\max}^{\omega} - (\epsilon_i^{\omega} \cdot \rho_i)) - \sum_{i \in V} \sum_{j \in \text{Succ}(i)} \sum_{t \in W_{ij}} (v_{ijt}^{\omega} \cdot \eta_{ijt}) \\
& - \sum_{k \in \mathcal{R}} \sum_{t=0, \dots, T} (\theta_{kt} \cdot \zeta_{kt}^{\omega}) \cdot x_{\omega} + \sum_{i \in V} \rho_i \\
& + \sum_{i \in V} \sum_{j \in \text{Succ}(i)} \sum_{t \in W_{ij}} \eta_{ijt} + \sum_{k \in \mathcal{R}} \sum_{t=0, \dots, T} \theta_{kt} \quad (26.54)
\end{aligned}$$

For given values  $(\rho, \eta, \theta)$  of Lagrangian multipliers, the associated dual function  $L(\rho, \eta, \theta)$  can be computed solving the following Lagrangian sub-problem:

$$Z[LSP(\rho, \eta, \theta, x)]: \text{Min. } \sum_{\omega \in \Omega} (C_{\max}'^{\omega} \cdot x_{\omega}) \quad (26.55)$$

$$\text{s. t. } x_{\omega} \geq 0 \quad (\omega \in \Omega) \quad (26.56)$$

where

$$C_{\max}'^{\omega} = C_{\max}^{\omega} - \epsilon_i^{\omega} \cdot \rho_{i(\omega)} - \sum_{i \in V} \sum_{j \in \text{Succ}(i)} \sum_{t \in W_{ij}} (v_{ijt}^{\omega} \cdot \eta_{ijt}) - \sum_{k \in \mathcal{R}} \sum_{t=0, \dots, T} (\theta_{kt} \cdot \zeta_{kt}^{\omega}) \quad (26.57)$$

Hence, obviously we have:

$$\begin{aligned}
Z[L(\rho, \eta, \theta)] = & Z[LSP(\rho, \eta, \theta)] + \sum_{i \in V} \rho_i \\
& + \sum_{i \in V} \sum_{j \in \text{Succ}(i)} \sum_{t \in W_{ij}} \eta_{ijt} + \sum_{k \in \mathcal{R}} \sum_{t=0, \dots, T} \theta_{kt} \quad (26.58)
\end{aligned}$$

The Lagrangian sub-problem ( $LSP(\rho, \eta, \theta)$ ) can be solved to optimality by setting  $\bar{x}_{\omega}$  equal to 1 if  $C_{\max}'^{\omega} < 0$ , or equal to 0 otherwise.

$Z[L(\rho, \eta, \theta)]$  defines a lower bound on the  $RMP_2$ , given that each feasible solution for the original problem is also feasible for the Lagrangian function. Hence, we can obtain the best possible lower bound by solving the Lagrangian dual problem ( $LDRMP_2$ ):

$$Z[LDRMP_2] = \text{Max}_{\rho, \eta, \theta} L(\rho, \eta, \theta) \quad (26.59)$$



As was done for the first proposed Lagrangian model (see Sect. 26.4.1) we use the subgradient procedure for estimating the values of the Lagrangian multipliers  $(\rho, \eta, \theta)$ . Hence, after solving  $LSP(\rho, \eta, \theta)$  we obtain the column assignment vector  $\bar{x}$ . Consequently, we can define a subgradient for  $L(\rho, \eta, \theta)$  as follows:

$$\psi_i = 1 - \sum_{\omega \in \Omega} (x_\omega \cdot \epsilon_i^\omega) \quad (i \in V) \quad (26.60)$$

$$\vartheta_{ijt} = 1 - \sum_{\omega \in \Omega} (x_\omega \cdot v_{ijt}^\omega) \quad (i \in V; j \in Succ(i); t \in W_{ij}) \quad (26.61)$$

$$\varphi_{kt} = 1 - \sum_{\omega \in \Omega} (x_\omega \cdot \zeta_{kt}^\omega) \quad (k \in \mathcal{R}; t = 0, \dots, T) \quad (26.62)$$

Now, given an upper bound  $T$  for the makespan and a step size  $sp$ , we can update the current Lagrangian multipliers by:

$$\rho_i = \rho_i + (sp \cdot \psi_i) \quad (i \in V) \quad (26.63)$$

$$\eta_{ijt} = \min\{0, \eta_{ijt} + (sp \cdot \vartheta_{ijt})\} \quad (i \in V; j \in Succ(i); t \in W_{ij}) \quad (26.64)$$

$$\theta_{kt} = \min\{0, \theta_{kt} + (sp \cdot \varphi_{kt})\} \quad (k \in \mathcal{R}; t = 0, \dots, T) \quad (26.65)$$

Thereafter, we remind that the step size  $sp$  is defined as follows:

$$sp = vp \cdot \frac{(T - L(\rho, \eta, \theta))}{Norm} \quad (26.66)$$

where  $Norm$  is given by:

$$Norm = \sum_{i \in V} (\psi_i)^2 + \sum_{i \in V} \sum_{j \in Succ(i)} \sum_{t \in W_{ij}} (\vartheta_{ijt})^2 + \sum_{k \in \mathcal{R}} \sum_{t=0, \dots, T} (\varphi_{kt})^2 \quad (26.67)$$

As was explained in Sect. 26.4.1 we start the subgradient procedure by setting  $vp = 2$ . At the end of each iteration we update the parameter  $vp$  with a systematic geometric revision:  $vp = gp \cdot vp$ . Normally the second parameter  $gp$  ranges between 0.87 and 0.9995, depending on the targeting problem.

The Lagrangian multipliers are updated during a limited number of iterations. Hence, we use the last updated multipliers as an estimation of the dual multipliers for pricing out new columns. Therefore, as was done in Sect. 26.4.1 for solving  $RMP_1$ , when there are no more columns with a negative reduced cost by using the Lagrangian multipliers, we perform a fixed number of iterations solving the  $RMP_2$  with the simplex method by using the solver for obtaining the values of the dual variables for pricing out new columns. Hence, if after the fixed number of iterations there are still columns with negative reduced costs, we go back to the Lagrangian procedure, otherwise, we stop.

## 26.5 Columns Initialization and Selection

Finally, before reporting the obtained computational experiments, we describe how the subset of columns  $\bar{\Omega}$  is initialized for the first CG iteration. In addition, we introduce the procedure developed for selecting the columns that should be included in the pool of activity work patterns in each CG iteration.

### 26.5.1 Columns Initialization

For the first CG iteration, we initialize the subset of columns  $\bar{\Omega}$  for solving the RMP( $\bar{\Omega}$ ) according to a schedule obtained by the tabu search (TS) developed by Bellenguez-Morineau (2008). Nevertheless, it is important to state that a reformulation that includes the utilization of slack variables in the models proposed in Sects. 26.3.1.1 and 26.3.1.2 allows solving the RMP without having an initial set of columns  $\bar{\Omega}$ . Thereafter, preliminary results show that initializing the pool of columns by means of the TS allows us to prove optimality faster and enhance the possibility of keeping a structure of activity work patterns that could lead to an integer feasible schedule.

### 26.5.2 Columns Selection

As pointed out previously, exhibiting an activity pattern with global minimal reduced cost requires the enumeration of each activity  $i$  for each time instant of its starting domain ( $ES_i \leq t_i \leq LS_i$ ). To limit the number of sub-problems solved at each iteration of the CG procedure, we propose filtering procedures ensuring that the considered pairs  $(i, t)$  may lead to columns with a negative reduced cost ( $rc_{it} < 0$ ).

First, according to duality properties, simplex multipliers  $\theta_{kt}$  associated with constraints (26.4) and (26.14) for  $RMP_1$  and  $RMP_2$ , respectively, are less than or equal to zero. Consequently, we have necessarily from (26.10) and (26.19) that  $rc_{it}^2 \geq 0$ . Thus, the column associated with the pair  $(i, t)$  might have a negative reduced cost only if  $rc_{it}^1 < 0$ . Clearly, the sub-problem  $SP(i, t)$  has to be solved only if this condition holds.

Then, assuming that  $lr_{it}^2$  is a lower bound on  $rc_{it}^2$ , we get  $rc_{it} \geq rc_{it}^1 + lr_{it}^2 = lr_{it}$ . Obviously, only sub-problems  $SP(i, t)$  for which  $lr_{it} < 0$  holds have to be considered since they potentially may lead to activity work patterns with a negative reduced cost. For a given pair  $(i, t)$ , such a lower bound  $lr_{it}^2$  can be computed by summing the  $\sum_{l \in \mathcal{L}} r_{il}$  smallest assignment costs  $c_{kl}$  among the workers that master at least one of the required skills to perform  $i$ .

Notice that intensive computational experiments reveal that adding several patterns to the current pool of columns at each iteration of the CG procedure leads

to better average CPU times. In most cases, the potential increasing in CPU time is counterbalanced by a decrease in the number of iterations needed for ensuring the convergence of the CG process.

## 26.6 Computational Results

Computational experiments were performed using the solver Gurobi Optimizer version 4.6. We selected a subset of the available instances for the MSPSP (Bellenguez-Morineau and Néron 2005) according to their size in terms of number of activities, skills, and number of workers. In general terms, the computational results shown in this section correspond to instances which contain between 20 and 62 activities, 2 and 15 skills, and 2 and 19 workers. We report results for 271 instances, which are divided into three groups:

- Group 1: We studied 110 instances from this group, containing between 20 and 51 activities, between 2 and 8 skills, and between 5 and 14 workers.
- Group 2: We included the results for 71 instances which contain between 32 and 62 activities, 9 and 15 skills, and 5 and 19 workers.
- Group 3: We studied 90 instances which contain between 22 and 32 activities, 3 and 12 skills, and 4 and 15 workers.

In Table 26.1 we compare the performance of the different column generation and Lagrangian relaxation approaches introduced in the previous sections. On one hand, we evaluated the CG approach based on the  $RMP_1$  (see Sect. 26.3.1.1) and using the simplex method for solving the linear program (LP). For notation purposes, we refer to this last approach as  $CG_1$ . In addition, we have also  $CGLR_1$ , in which the LP for the related  $RMP_1$  is solved with the combined Lagrangian relaxation and column generation approach proposed in Sect. 26.4.2. On the other hand, we have  $CG_2$  and  $CGLR_2$ , which correspond to the utilization of  $RMP_2$ , which is based on the master problem reformulation introduced in Sect. 26.3.1.1. Hence, in  $CG_2$  we use only the simplex method for solving the LP, while in  $CGLR_2$  we combine the use of Lagrangian relaxation and the simplex method for solving the LP, as it was proposed in Sect. 26.4.2.

Furthermore, in Table 26.1 we compare the results of the four mentioned approaches in terms of the average deviation  $\Delta_{LB^*}^\theta$  against the best known lower bounds ( $LB^*$ ) obtained by Bellenguez-Morineau and Néron (2005). Subsequently, we also compare the average computation times and the average number of generated columns obtained by each tested model for reaching their respective lower bound. It is important to mention that, for enforcing the quality of the lower bound obtained by applying column generation, we calculated a preliminary lower bound based on the principle of the stable set. This bound was proposed by Mingozzi et al. (1998) and adapted to the MSPSP by Bellenguez-Morineau and Néron (2005).

After analyzing the obtained results we can see that  $CG_2$  and  $CGLR_2$  lead to better lower bounds than  $CG_1$  and  $CGLR_1$ , implying that the solution of

**Table 26.1** Comparison between CG approaches proposed

		Group of instances		
		Group 1	Group 2	Group 3
Average deviation against $LB^*$	$CG_1$	-10.80 %	-4.96 %	-6.17 %
	$CGLR_1$	-10.80 %	-4.96 %	-6.17 %
	$CG_2$	-4.31 %	-3.20 %	-3.30 %
	$CGLR_2$	-4.31 %	-3.20 %	-3.30 %
Average CPU time (sec)	$CG_1$	11.37	9.88	5.10
	$CGLR_1$	7.07	7.97	3.42
	$CG_2$	216.58	119.82	95.98
	$CGLR_2$	193.15	99.54	87.37
Average number of generated columns	$CG_1$	738.05	1,181.65	1,817.77
	$CGLR_1$	1,181.19	1,645.07	2,571.80
	$CG_2$	3,498.45	3,841.40	5,814.11
	$CGLR_2$	9,336.71	10,370.57	11,458.41

$RMP_2$  indeed enhances a stronger linear relaxation than  $RMP_1$ . Nevertheless, the approaches based on the solution of  $RMP_2$  required a considerably larger amount of computation time until obtaining a lower bound. In addition, we can indeed notice that the utilization of Lagrangian relaxation allowed us to accelerate the solution of the respective restricted master problems. Moreover, we can see that the approaches based on the solution of  $RMP_2$  generates more columns (i.e., activity work patterns) per instance than  $CG_1$  and  $CGLR_1$ . Additionally, we can see that the number of generated columns also increases when using the approach that combines Lagrangian relaxation and the simplex method for solving the LP.

## 26.7 Conclusions

The main differences between the proposed approaches relies in the master problem formulation and the methods used for solving the related LP. Hence, we were able to conclude that  $RMP_2$  allowed us to obtain better linear relaxations than the ones obtained when using  $RMP_1$ . Nevertheless, we could argue that the improvement in the quality of the resulting lower bound after applying the  $RMP_2$  based approaches is not that significant, given the considerable increase in the computation time invested in the solution of each tested instance. Additionally, we were also able to establish that the utilization of the simplex method along with the proposed Lagrangian relaxation models allowed us to decrease the computation time consumed in the solution of each tested instance. Nevertheless, it is important to mention that there are some new perspectives that could be considered regarding to the utilization of column generation for solving the MSPSP. On one hand the generation of certain additional inequalities (cuts) could lead to a stronger linear relaxation when solving

the restricted master problem. In addition, regarding the particular performance of the  $RMP_2$  based approaches it could be interesting to take into account certain measures for accelerating the convergence, which could lead to a decrease in the computation times.

## References

- Artigues C, Demassez S, Néron E, (2008) Resource-constrained project scheduling: models, algorithms, extensions and applications. Wiley, Hoboken
- Barahona F, Anbil R (2000) The volume algorithm: producing primal solutions with a subgradient method. *Math Program* 87:385–399
- Barahona F, Jensen D (1998) Plant location with minimum inventory. *Math Program* 83:101–111
- Bard J, Purnomo H (2005) Preference scheduling for nurses using column generation. *Eur J Oper Res* 164:510–534
- Barnhart C, Johnson E, Nemhauser G, Savelsbergh M, Vance P (1998) Branch-and-price: column generation for solving huge integer programs. *Oper Res* 46:316–329
- Beliën J, Demeulemeester E (2007) On the trade-off between staff-decomposed and activity-decomposed column generation for a staff scheduling problem. *Ann Oper Res* 155:143–166
- Bellenguez-Morineau O (2008) Methods to solve multi-skill project scheduling problem. *4OR-Q J Oper Res* 6:85–88
- Bellenguez-Morineau O, Néron E (2005) Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills. In: Practice and theory of automated timetabling V. Lecture notes in computer science, vol 3616. Springer, Berlin, pp 229–243
- Bertsekas D (1999) Nonlinear programming. Athena Scientific, Belmont
- Bixby R, Gregory J, Lustig I, Marsten R, Shanno D (1992) Very large-scale linear programming: a case study in combining interior point and simplex methods. *Oper Res* 40:885–897
- Brucker P, Knust S (2000) A linear programming and constraint propagation-based lower bound for the RCPSP. *Eur J Oper Res* 127:355–362
- Brucker P, Drexel A, Möhring R, Neumann K, Pesch E (1999) Resource-constrained project scheduling: notation, classification, models, and methods. *Eur J Oper Res* 112:3–41
- Busacker R, Gowen P (1961) A procedure for determining a family of minimum-cost-flow patterns. Technical report 15, operations research office, Johns Hopkins University, Baltimore, MD
- Chen Z, Powell W (1999) A column generation based decomposition algorithm for a parallel machine just-in-time scheduling problem. *Eur J Oper Res* 116:220–232
- Cordeau J, Laporte G, Pasin F, Ropke S (2010) Scheduling technicians and tasks in a telecommunications company. *J Sched* 13:1–17
- Correia I, Lourenço L, Saldanha-da Gama F (2012) Project scheduling with flexible resources: formulation and inequalities. *OR Spectr* 34:635–663
- Dantzig G, Wolfe P (1960) Decomposition principle for linear programs. *Oper Res* 8:101–111
- Dohn A, Kolind E, Clausen J (2009) The manpower allocation problem with time windows and job-teaming constraints: a branch-and-price approach. *Comput Oper Res* 36:1145–1157
- Fábián C (2000) Bundle-type methods for inexact data. *Cent Eur J Oper Res* 8:35–55
- Firat M, Hurkens C (2012) An improved MIP-based approach for a multi-skill workforce scheduling problem. *J Sched* 15:363–380
- Gélinas S, Soumis F (2005) Dantzig-Wolfe decomposition for job shop scheduling. In: Desaulniers G, Desrosiers J, Solomon MM (eds) Column generation. Springer, New York, pp 271–302
- Gilmore P, Gomory R (1961) A linear programming approach to the cutting-stock problem. *Oper Res* 9:849–859
- Goffin J, Vial J (2002) Convex nondifferentiable optimization: a survey focused on the analytic center cutting plane method. *Optim Method Softw* 17:805–867

- Gutjahr W, Katzensteiner S, Reiter P, Stummer C, Denk M (2008) Competence-driven project portfolio selection, scheduling and staff assignment. *Cent Eur J Oper Re* 16:281–306
- Hartmann S, Briskorn D (2010) A survey of variants and extensions of the resource-constrained project scheduling problem. *Eur J Oper Res* 207:1–14
- Heimerl C, Kolisch R (2010) Scheduling and staffing multiple projects with a multi-skilled workforce. *OR Spectr* 32:343–368
- Held M, Karp R (1971) The traveling-salesman problem and minimum spanning trees: part II. *Math Program* 1:6–25
- Held M, Wolfe P, Crowder H (1974) Validation of subgradient optimization. *Math Program* 6: 62–88
- Huisman D, Jans R, Peeters M, Wagelmans A (2005) Combining column generation and lagrangian relaxation. In: Desaulniers G, Desrosiers J, Solomon MM (eds) *Column generation*. Springer, New York, pp 247–270
- Ioachim I, Desrosiers J, Soumis F, Belanger N (1999) Fleet assignment and routing with schedule synchronization constraints. *Eur J Oper Res* 119:75–90
- Jans R, Degraeve Z (2004) An industrial extension of the discrete lot-sizing and scheduling problem. *IIE Trans* 36:47–58
- Jaumard B, Semet F, Vovor T (1998) A generalized linear programming model for nurse scheduling. *Eur J Oper Res* 107:1–18
- Jiang H, Krishnamoorthy M, Sier D (2004) Staff scheduling and rostering: theory and applications, part I and II. *Ann Oper Res* 128:1–4
- Li H, Womer K (2009) Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm. *J Sched* 12:281–298
- Lübbecke M, Desrosiers J (2005) Selected topics in column generation. *Oper Res* 53:1007–1023
- Mason A, Smith M (1998) A nested column generator for solving rostering problems with integer programming. In: *Proceedings of international conference on optimisation: techniques and applications*, pp 827–834
- Mehrotra A, Murphy K, Trick M (2000) Optimal shift scheduling: a branch-and-price approach. *Nav Res Log* 47:185–200
- Mingozzi A, Maniezzo V, Ricciardelli S, Bianco L (1998) An exact algorithm for the resource-constrained project scheduling problem based on a new mathematical formulation. *Manag Sci* 44:714–729
- Montoya C, Bellenguez-Morineau O, Rivreau D (2013) Branch and price approach for the multi-skill project scheduling problem. *Optim Lett*. Doi:10.1007/s11590-013-0692-8
- Polyak B (1967) A general method of solving extremum problems. *Sov Math Doklady* 8:593–597
- Van den Akker J, Hurkens C, Savelsbergh M (2000) Time-indexed formulations for machine scheduling problems: column generation. *INFORMS J Comput* 12:111–124
- Van den Akker J, Hoogeveen J, de Velde S (2002) Combining column generation and Lagrangian relaxation to solve a single-machine common due date problem. *INFORMS J Comput* 14: 37–51
- Van den Akker J, Diepen G, Hoogeveen J (2007) A column generation based destructive lower bound for resource constrained project scheduling problems. In: *Integration of AI and OR techniques in constraint programming for combinatorial optimization problems (CPAIOR 2007)*. Lecture notes in computer science, vol 4510. Springer, Berlin, pp 376–390
- Wolsey L (1998) *Integer programming*. Wiley, New York