Computers & Operations Research 52 (2014) 282–290

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Variable neighborhood search for the workload balancing problem in service enterprises

Thanh-Ha Nguyen*, Mike Wright

Lancaster University Management School, Lancaster LA1 4YX, UK

ARTICLE INFO

Available online 8 August 2013

Keywords:
Scheduling
Quadratic integer programming
Variable neighborhood search
Meta-heuristic
Capacity management

ABSTRACT

In this paper, we consider a telecommunication service company facing seasonal demand and time-varying capacity. A uniform lead-time, which is the maximum time span a customer has to wait before receiving the required service, is quoted to all customers. We present a quadratic integer programming model for the problem of scheduling jobs to meet the promised lead-time with the objective of balancing the workload across time. Since in practice solving such a problem to optimality can be very difficult, two variants of a variable neighborhood search approach are proposed. Extensive computational tests show that our heuristics are able to provide high quality solutions efficiently.

1. Introduction

In the telecommunications industry, a common practice is to use uniform lead time, which is the maximum time span a customer has to wait before receiving the required service. For services (e.g. installation of broadband at telecom exchanges) where customers do not have to participate in the service delivery process, the advantages of quoting a uniform lead time are twofold. On the one hand, all customers are guaranteed a uniform delivery lead time. On the other hand, the firm can flexibly choose the best time within the promised lead-time to carry out the work. We consider a telecommunication operator with seasonal demand for such a service. The available capacity per period also follows a cyclic pattern. In order to improve service efficiency, i.e. to avoid unnecessary capacity over-utilization in certain periods and under-utilization in other periods, the workload must be balanced across time. Thus, we are interested in how to match the firm's capacity to customer demand in such a way that: (a) the quoted lead-time is satisfied and (b) the capacity utilization rate per period is distributed as equally as possible along the time line. This problem will be referred to as the Workload Balancing Problem (WBP) in the rest of the paper.

A clear definition for the measure of balance is necessary for the study of workload balancing. Naturally, variance, standard deviation or *sum of squared deviations* (*SSD*) are good measures, as they tend to penalize larger deviations at a higher rate. We model the WBP in terms of minimizing the SSD of capacity utilization rates across time periods from their mean. The WBP can be seen as a Quadratic Integer Programming Problem (QIP or QIPP) due to the formulation of the problem as an integer problem with a quadratic objective function. The QIP is, in general, difficult to solve. Hence mathematical programming techniques may fail to deliver an exact solution in reasonable time. To practically solve the QIP, heuristic algorithms which find high quality solutions in short computation time have been proposed. Such heuristic algorithms are variable neighborhood search [1–3], simulated annealing [4–6], tabu search [7–9], genetic algorithms [10–14], evolution strategies [15,16], ant algorithms [17–20], and scatter search [21,12]. Among them, variable neighborhood search (VNS) is a simple and powerful search method for solving combinatorial problems. VNS-based heuristics are reported to be among the best performing algorithms for a number of problems such as the Traveling Salesman Problem (TSP) [3] and the Quadratic Assignment Problem (QAP) [22]. The latter is NP-hard and is regarded as one of the hardest QIP [23]. The QAP can be described as the optimization problem of assigning a set of facilities to a set of locations with given distances between the locations and given flows between the facilities in order to minimize the sum of the product between flows and distances. For solving the QAP, Stützle presented in [22] heuristic algorithms with perturbation operator considering change of the neighborhood in the search, which can be seen as variants of VNS. His test results show that those VNS-based heuristics have excellent performance when compared to robust tabu search and MAX–MIN ant system, which are known to perform well for the QAP [18,24]. For this reason, we propose two variants of a VNS method to tackle the WBP.

The WBP was only once investigated by Li and He [25]. They proposed two greedy local search algorithms to tackle the problem. However, these algorithms are time consuming and because

* Corresponding author. Tel.: +49 176 56375220.
E-mail addresses: n.thanh-ha@lancaster.ac.uk (T.-H. Nguyen), m.wright@lancaster.ac.uk (M. Wright).

0305-0548/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cor.2013.07.027

of the greedy strategy, they can often get stuck in the local optima. Another problem is that without providing a mathematical model and thus an exact method, in their experiments, the previous authors were only able to compare CPU times, but do not obtain any information about solution quality in terms of loss of optimality.

This paper proposes several improvements to the previously published work by Li and He [25] as follows: Section 2 presents a mathematical model for the WBP, based on which an exact solution can be derived. In Section 3, besides an exact method, two new heuristics based on VNS are proposed to improve search speed and solution quality. In Section 4, to illustrate the proposed heuristic algorithms, a concrete numerical example is provided. Section 5 shows computation results for 140 test problem instances. Section 6 concludes and discusses potential extensions and directions for further research.

2. Problem formulation

2.1. Assumptions

We consider a major telecommunication service company. The company is using an integrated planning system, based on hierarchical planning concepts that allow one to decompose the entire planning problem into partial planning tasks but to still consider their interdependencies and to coordinate their solutions. This planning system consists of different modules, such as demand forecasting, resources planning, work scheduling, which are interlinked with each other. It makes use of solution approaches known as mathematical programming and meta-heuristics and provides supports at different levels for planning tasks along the company's service chain, from long-term strategic decision making to short-term operational decisions. The levels for planning may overlap or may be distinct. Either way there is a flow of information from strategic to operational planning and then to operational planning and vice versa.

Demand: The firm faces seasonal demand for a particular service, e.g. broadband installation. The estimated demand data are provided by the responsible module for forecasting demand. Thus, the start and the end point of the seasonal cycle are known. Further, the seasonal demand pattern, which repeats itself for every cycle, is also given. By dividing the seasonal cycle into time periods $1, \ldots, M$, the demand pattern can be expressed by a vector $[\lambda_1, \ldots, \lambda_M]^T$ (See Table 1 for the notations used). Each element of this vector represents the demand (measured by the number of jobs) that occurs through a particular time period.

Table 1
Notation for problem formulation.

M	Length of one demand-cycle
N	Length of one capacity-cycle
τ	Finite planning horizon, $\tau = lcm(M, N)$
I	Set of job arrival dates, $I = \{1, \ldots, \tau\}, i \in I$
J	Set of job completion dates, $J = \{1, \ldots, \tau\}, j \in J$
J_i	Set of feasible completion dates of jobs arriving in period i
λ_i	Demand in period i
c_j	Available capacity in period j
x_{ij}	Number of jobs that arrive at period i and are assigned to completion date j
X	Assignment scheme $X = [x_{ij}]_{\tau \times \tau}$
ℓ	The minimum time span a job has to wait before it can be processed
L	Uniform lead-time, $\ell < L \le \tau + \ell$
u_j	Used capacity in period j
μ_j	Capacity utilization rate in period j
$\overline{\mu}$	Mean of capacity utilization rates during the planning horizon τ
r_i	Internal release date of jobs that arrive in period i
d_i	Due date of jobs that arrive in period i

Capacity: The firm has a fixed number of permanent employees and a number of seasonal technicians with repeated fixed term contracts. The latter are retained in order to meet peaks in demand (e.g. surge of demand for broadband installations at the beginning of school terms). The information concerning availability of the workforce per time period is provided by means of the medium-term, anticipatory deployment plan. As it is possible to estimate the average time a technician needs to complete a job, we represent capacity during a time period in terms of the number of jobs to better match it with customer demand. Capacity levels are assumed to follow a cycle of N time periods with the pattern $[c_1, \ldots, c_N]^T$.

Planning horizon: The planning horizon τ is the minimum time interval after each of which both demand and capacity pattern will repeat themselves. Thus, τ is determined as the least common multiplier of M and N, $\tau = lcm(M, N)$.

Lead time: Uniform lead-time L is quoted to all customers. We assume that L is bounded by $\ell+1$ and $\tau+\ell$ ($\ell < L \le \tau+\ell$), where ℓ denotes the minimal time span a job has to wait before it can be processed.

Job: All jobs are the same and can be completed within one time period. Each job is characterized by its arrival i and the time period j, when it is completed.

For the ease of notation, let I and J where $I = J = \{1, \ldots, \tau\}$ denote the set of job arrival dates and job completion dates respectively. The demand and the available capacity during the planning horizon are represented by the vectors $\lambda = [\lambda_i]$ and $c = [c_j]$, where $i \in I$ and $j \in J$. The vector λ is obtained by τ/M-times concatenation of $[\lambda_1, \ldots, \lambda_M]^T$, and the vector c by τ/N-times concatenation of $[c_1, \ldots, c_N]^T$.

We want to find a job assignment scheme $X = [x_{ij}]_{\tau \times \tau}$ that determines how many jobs of each demand λ_i are to be completed in time period j, so that all demands are met within L periods and the workload is balanced over time. In an ideal case, the capacity utilization rate in every time period would be the same.

In the following, the lead-time constraints and the objective function of the WBP are specified. Note that, an explicit formulation of capacity constraints is not needed. The reasons are as follows: (a) If the total capacity is sufficient to accommodate the total demand over τ periods (that is, $\sum_{i \in I} \lambda_i \le \sum_{j \in J} c_j$) and the quoted lead-time L is long enough, the objective function will ensure that workload is distributed across periods as equally as possible and the capacity utilization rate in each period is less than or equal to 100%. (b) Otherwise, the firm will not be able to accommodate the total demand within the promised lead-time without exceeding the total capacity. In this case, the firm is better off increasing capacity by hiring more labor or outsourcing part of its service activities, but modeling these aspects is out of the scope of this paper.

2.2. Lead-time constraints

We use internal release and due dates to indicate the time window in which a job must be completed. The internal release date is the earliest possible time a job can start. The internal release date $r_i \in \{1, \ldots, \tau\}$ of jobs, that arrive in period i, is computed as

$$r_i = (i + \ell + 1) \bmod \tau \tag{1}$$

The modulo operation (mod τ) is used here and in the rest of the paper to model the cyclical behavior of demand and capacity that recurs every τ period. The due date $d_i \in \{1, \ldots, \tau\}$ of jobs arriving in period i is the latest time these jobs must be completed, and is calculated as arrival time plus quoted lead-time.

$$d_i = (i + L) \bmod \tau \tag{2}$$