Discrete Applied Mathematics 182 (2015) 115–121

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

A pseudo-polynomial time algorithm for solving the resource dependent assignment problem

Dvir Shabtay [a], George Steiner [b,*], Liron Yedidsion [c]

[a] Department of Industrial Engineering and Management, Ben-Gurion University of the Negev Beer-Sheva, Israel

[b] Operations Management Area, DeGroote School of Business, McMaster University, Hamilton, Ontario, Canada

[c] Faculty of Industrial Engineering and Management, Technion - Israel Institute of Technology, Haifa, Israel

ARTICLE INFO

Article history:
Received 15 January 2013
Received in revised form 18 July 2013
Accepted 23 August 2013
Available online 19 September 2013

Keywords:
Assignment problem
Pseudo-polynomial time algorithm
Resource allocation
Bicriteria optimization

ABSTRACT

In this paper the resource dependent assignment problem (*RDAP*) is considered. In the *RDAP* the cost of assigning agent j to task i is a multiplication of task i's cost parameter by a cost function of agent j and the cost function of agent j is a linear function of the amount of resource allocated to the agent. A solution for the *RDAP* problem is defined by the assignment of agents to tasks and by a resource allocation to each agent. The quality of a solution is measured by two criteria. The first criterion is the total assignment cost and the second one is the total weighted resource consumption. Yedidsion et al. showed that the bicriteria variations of the problem are all $\mathcal{NP}$-hard for *any* given set of task costs. However, whether these problems are strongly or ordinarily $\mathcal{NP}$-hard remained an open question. In this paper we close this gap by providing pseudo-polynomial time algorithms for solving these problems.

1. Introduction

Yedidsion et al. [4] have presented and analyzed a new assignment problem which they refer to as the *resource dependent assignment problem* or *RDAP* in short. The problem can be presented as follows: a set of n *agents* is to be assigned to a set of n *tasks* such that each task is performed only once and each agent is assigned to a single task. The cost of assigning agent j to task i is given by

$$c_{ij} = \omega_i \times p_j\left(u_j\right), \tag{1}$$

where ω_i is task i's *assignment cost parameter* and $p_j(u_j)$ is the *assignment cost function* of agent j. The assignment cost function is given by the following linear model for $j = 1, \ldots, n$:

$$p_j\left(u_j\right) = \overline{p}_j - b_j u_j, \quad 0 \le u_j \le \overline{u}_j < \overline{p}_j / b_j, \tag{2}$$

where $\overline{p}_j$ is the non-compressed (maximum) assignment cost for agent j; u_j is a decision variable that represents the amount of a nonrenewable resource allocated to agent j; $\overline{u}_j$ is the upper bound on the amount of resource that can be allocated to agent j; and b_j is the positive cost compression rate of agent j. In this paper, we assume that the resource is divisible and thus can be used in *continuous* quantities.

A solution for the RDAP problem is defined by a permutation

$$\phi = (\phi(1), \phi(2), \ldots, \phi(n)),$$

* Corresponding author. Tel.: +1 905 525 9140; fax: +1 905 521 8995.
E-mail addresses: dvirs@bgu.ac.il (D. Shabtay), steiner@mcmaster.ca (G. Steiner), lirony@ie.technion.ac.il (L. Yedidsion).

0166-218X/$ – see front matter
http://dx.doi.org/10.1016/j.dam.2013.08.037

where $j = \phi(i)$ means that agent j is assigned to task i in permutation ϕ, and by a resource allocation vector $\mathbf{u} = (u_1, u_2, \ldots, u_n)$. The quality of a solution is measured by two different criteria. The first is the *total assignment cost* defined by

$$c(A) = \sum_{i=1}^{n} \omega_i \times (\overline{p}_{\phi(i)} - b_{\phi(i)} u_{\phi(i)}) = \sum_{i=1}^{n} c_{i,\phi(i)}(u_{\phi(i)}), \tag{3}$$

where $A = (\phi, \mathbf{u})$. The second criterion is the *total resource consumption cost*, given by

$$U(A) = \sum_{j=1}^{n} v_j u_j, \tag{4}$$

where v_j is the cost of assigning one unit of resource to agent j. Both criteria have to be minimized.

Yedidsion et al. [4] showed that the *RDAP* has many real life applications. Among those applications is a large set of scheduling problems with controllable processing times in which their scheduling criterion can be represented by or reduced to the format in (3). They studied the following four variants of the *RDAP*.

- The first one, denoted by *RDAP*1, is to minimize the total integrated cost, $c(A) + U(A)$ defined by

  $$\sum_{i=1}^{n} \omega_i \left(\overline{p}_{\phi(i)} - b_{\phi(i)} u_{\phi(i)}\right) + \sum_{i=1}^{n} v_{\phi(i)} u_{\phi(i)}$$

  subject to

  $$0 \le u_j \le \overline{u}_j \quad \text{for } j = 1, \ldots, n. \tag{5}$$
- The second problem, denoted by *RDAP*2, is to minimize $c(A)$ given by Eq. (3) subject to Eq. (5) and

  $$U(A) = \sum_{j=1}^{n} v_j u_j \le U_v, \tag{6}$$

  where U_v is an upper bound on the total resource consumption cost (or budget).
- The third version, denoted by *RDAP*3, is to minimize Eq. (4) subject to Eq. (5) and

  $$c(A) \le K, \tag{7}$$

  where K is a given upper bound on the total assignment cost.
- The last one, denoted by *RDAP*4, is to identify a *Pareto-optimal* solution for each Pareto-optimal point, where a solution A with $c = c(A)$ and $U = U(A)$ is called *Pareto-optimal* (or *efficient*) if there does not exist another solution A' such that $c(A') \le c(A)$ and $U(A') \le U(A)$ with at least one of these inequalities being strict. The corresponding Pareto-optimal point is (c, U).

It should be noted that solving *RDAP*4 also solves *RDAP*1–*RDAP*3 as a by-product.

Yedidsion et al. [4] showed that *RDAP*1 can be reduced to the classical *linear assignment problem* (*LAP*) and thus can be solved in $O(n^3)$ time. Furthermore, by using a reduction from the *Partition* problem, they proved that *RDAP*2–*RDAP*4 are all $\mathcal{NP}$-hard for *any* given set of task cost parameters with $\omega_i \neq \omega_j$ for any $i \neq j$ even if $v_j = 1$ for $j = 1, \ldots, n$. However, the question whether *RDAP*2–*RDAP*4 are strongly or ordinarily $\mathcal{NP}$-hard remained open. In this paper, we close this gap in the literature by providing pseudo-polynomial time algorithms for solving *RDAP*2–*RDAP*4 problems. The algorithms are based on the crucial insight that for each Pareto point there exists an efficient solution A with the assignment cost function of all but at most one agent either fully reduced, that is $p_j\left(u_j\right) = \overline{p}_j - b_j \overline{u}_j$, or not reduced at all, that is, $p_j\left(u_j\right) = \overline{p}_j$. This property of an efficient solution is known as the *all-or-none* property and is frequently observed in scheduling problems with controllable processing times (see, e.g., [3,2]).

The rest of the paper is organized as follows. In Section 2 we present some crucial properties of efficient solutions. In Section 3 we show how we can use these properties to construct a pseudo-polynomial time algorithm for the *RDAP*2 problem and in Section 4 we show how the pseudo-polynomial time algorithm for the *RDAP*2 problem can be converted to solve the *RDAP*3 and *RDAP*4 problems in pseudo-polynomial time as well. A summary and discussion section concludes the paper.

2. Properties of efficient solutions

Hereafter, without loss of generality, we assume that $\omega_1 \ge \omega_2 \ge \cdots \ge \omega_n \ge 0$. The following lemma taken from [4] can easily be derived from a well-known result in linear algebra regarding the minimization of a scalar product of two vectors (see Hardy et al. [1]).

Lemma 1. *For any given resource allocation vector* $\mathbf{u} = (u_1, u_2, \ldots, u_n)$*, which fixes the agent cost function, the optimal assignment,* ϕ^**, can be obtained in* $O(n \log n)$ *time for all problem variations by ordering components of the* $\mathbf{p} = (p_1(u_1), p_2(u_2), \ldots, p_n(u_n))$ *vector in a non-decreasing order. The optimal assignment is then attained by matching the agent in the jth position in this vector to task j.*