# Optimal Discounts for the Online Assignment Problem


Michael Z. Spivey

*Department of Mathematics and Computer Science, University of Puget Sound, Tacoma, WA 98416-1043 USA*



## Abstract

We prove that, for two simple functions $d_{rlt}$, solving the online assignment problem with $c_{rl} - d_{rlt}$ as the contribution for assigning resource $r$ to task $l$ at time $t$ gives the optimal solution to the corresponding offline assignment problem (provided the optimal offline solution is unique). We call such functions $d_{rlt}$ *optimal discount functions.*



*Keywords:* Assignment problem, Online assignment problem, Online matching, Linear programming


## 1. Introduction

Suppose we have a set of resources and a set of tasks that must be assigned to each other over time. All of the resources are available initially. The tasks arrive in an online fashion; that is, each task arrives singly over time. As each task $l$ arrives it is assigned to a resource $r$, resulting in a contribution $c_{rl}$, or not, in which case it leaves the network permanently. The objective is to maximize the sum of the assignment contributions. Problems of this kind are called *online*, or *dynamic, assignment problems*. Variations and different aspects of dynamic assignment problems have been studied; see, for example, [1, 2, 3, 4, 5].

The greedy online algorithm that chooses the best assignment at time $t$ without regard to the future tends to perform poorly when compared to the optimal offline solution. For instance, if costs in the minimization version of the problem are independent and identically distributed in $[0, 1]$, then the expected value of the greedy solution is $H_{n+1} - 1 \approx \log n$ [6, p. 145], where $H_n = \sum_{i=1}^{n} 1/i$, and so is unbounded. However, the expected value of the minimization version of the corresponding offline optimal solution is $\pi^2/6$ [7]; not only is it bounded, it is smaller than 2! On the other hand, Hoffman [8] found a condition that implies that a variation of the greedy online solution gives the optimal offline solution. If the contributions for the offline $n \times n$ maximization problem form an *anti-Monge sequence*; i.e., if $c_{ij} + c_{kl} \geq c_{il} + c_{kj}$ for all $1 \leq i < k \leq n$ and $1 \leq j < l \leq n$, then assigning resource $i$ to task $i$ for all $i$ is optimal.

An idea considered in previous work [9] by the author for addressing the typically poor quality of the greedy solution is to subtract from the contribution $c_{rl}$ for assigning resource $r$ to task $l$ at time $t$ a discount function $d_{rlt}$. Such a discount function attempts to balance the contribution gained from making an assignment now against the potential contribution from a better future assignment. Two of the discount functions we consider are $d_{rlt} = v_{r,t+1}$ and $d_{rlt} = v_{r,t+1} + v_{l,t+1}$. These functions can be thought of as discounting by the value of having resource $r$ or by the sum of the values of resource $r$ and task $l$, at time $t + 1$. (We formally define $v_{rt}$ and $v_{lt}$ in Section 2.) Such discounts should seem reasonable, especially in the first case; the time $t$ decision should be a tradeoff between the contribution $c_{rl}$ for assigning $r$ to the arriving task $l$ versus the value of having $r$ available in the next time period. (Task $l$ leaves the network if it is not assigned and so is not available at $t + 1$ anyway.)

However, calculating the value of having a resource or a task available at a future time requires knowledge of the future state of the network, which of course is not available at the time that a decision needs to be made. Discount functions such as these then really make sense in an adaptive setting, in which many instances of similar online assignment problems are solved over a long period of time. For example, consider the problem of a delivery company's assigning drivers to loads day after day. While the drivers and loads may not be exactly the same every day, they are often similar. In a setting such as this one, we can, after we know the full history of many problem instances, build an approximation of the value of a resource and of a task at time $t$ by considering the values of similar resources and tasks in problems already solved. Then we can use these approximate values from time $t + 1$ as discounts on the time $t$ decision in the current problem. This is the approach followed in our previous work.

How good are the discount functions $d_{rlt} = v_{r,t+1}$ and $d_{rlt} = v_{r,t+1} + v_{l,t+1}$, though? Numerical simulations [9] indicate that they give approximately optimal solutions in certain settings in which the resource and task properties are allowed to change from problem instance to problem instance. However, we were surprised to discover that when we considered exactly the same problem over more


*Email address:* `mspivey@pugetsound.edu` (Michael Z. Spivey)

than one iteration, discounting by resource gradients or by resource and task gradients produced the optimal offline solution rather than an approximation of it. This held true for every problem instance we considered. In other words, it appeared that if one were to have (e.g., build up over time) the correct resource and task values for a particular problem, then using those values as discounts would cause the greedy online solution to yield the optimal offline solution. Thus not only do $d_{rlt} = v_{r,t+1}$ and $d_{rlt} = v_{r,t+1} + v_{l,t+1}$ appear to be effective as discount functions, they actually look like what we might call *optimal discount functions*; i.e., discount functions that cause the online solution to be the same as the optimal offline solution. In our paper [10] we were able to prove that $d_{rlt} = v_{r,t+1}$ is such an optimal discount function, but we were unable to adapt the proof for the case of the resource and task gradients.

The main contributions of the current paper are twofold: 1) We prove that $d_{rlt} = v_{r,t+1} + v_{l,t+1}$ is also an optimal discount function. 2) This proof gives much more insight into why the two discount functions are optimal, partly because it is quite a bit simpler than the one in Spivey and Powell [10]. It immediately implies that $d_{rlt} = v_{r,t+1}$ is optimal as well.

(In [9] we consider a third discount function $d_{rlt} = v_{rl,t+1}$, which can be thought of as the value of having the resource and task together as a pair available at time $t+1$. While similar to $d_{rlt} = v_{r,t+1} + v_{l,t+1}$, experimental work shows that it cannot be an optimal discount function. It does, however, perform well in the adaptive setting we consider. For more on building approximations to the value of some resource, task, or even network state over time see work by George, Powell, and Kulkarni [11] or Powell's recent text [12] on approximate dynamic programming – especially Chapter 11.)

## 2. The discount functions

Let $R$ denote the set of resources and $L$ the set of tasks, with $R$ and $L$ disjoint. For our purposes we define the (offline) assignment problem as follows:

$$\begin{aligned}
\text{Maximize } & \sum_{r \in R} \sum_{l \in L} c_{rl} x_{rl} \\
\text{subject to } & \sum_{l \in L} x_{rl} \leq 1 \text{ for each } r \in R; \\
& \sum_{r \in R} x_{rl} \leq 1 \text{ for each } l \in L; \\
& x_{rl} \in \{0,1\} \text{ for each } r, l.
\end{aligned}$$

In the online version, all resources in $R$ are available initially, and we step through time, adding a task at each time $t$. If $R_t$ is the set of resources available at time $t$, $l$ is the task that becomes available at time $t$, and $d_{rlt}$ is the discount function on the contribution for assigning $r$ to $l$ at time $t$, our problem at time $t$ is to solve

$$\begin{aligned}
\text{Maximize } & \sum_{r \in R_t} (c_{rl} - d_{rlt}) x_{rlt} \\
\text{subject to } & \sum_{r \in R_t} x_{rlt} \leq 1; \\
& x_{rlt} \in \{0,1\} \text{ for each } r \in R_t.
\end{aligned}$$

In Spivey and Powell [9] we propose the discount functions $d_{rlt} = v_{r,t+1}$ (discounting with *resource gradients*) and $d_{rlt} = v_{r,t+1} + v_{l,t+1}$ (discounting with *resource* and *task gradients*). The quantity $v_{rt}$ is the marginal value of resource $r$ at time $t$, in the sense that if $r$ is available at time $t$, $v_{rt}$ is the net amount that would be lost by removing $r$ from the network, and if $r$ is not available at time $t$, $v_{rt}$ is the net amount that would be gained by adding $r$ to the network. Thus our two functions can be thought of as discounting the contribution for $c_{rlt}$ by either (1) the value of having both resource $r$ and task $l$ available at time $t+1$, or (2) the value of having just resource $r$ available at time $t+1$. This definition implies that we have an underlying solution $\mathbf{x}'$ from which to calculate these gradients. As we indicate in the Introduction, it is the optimal offline solution that we need for $\mathbf{x}'$.

Formally, we define $v_{rt}$ and $v_{lt}$ in the following fashion. Let $R'_t$ be the set of all resources that are available for assignment at time $t$, and let $L'_t$ be the set of all tasks that become available for assignment at time $t$ or later, under a particular solution $\mathbf{x}'$. (In other words, these resources and tasks have not been assigned by time $t$ under $\mathbf{x}'$.) Let $S_t = R'_t \cup L'_t$. Define $C(S_t)$ to be the value of the optimal assignment problem for $S_t$; i.e.,

$$\begin{aligned}
C(S_t) = \text{Max } & \sum_{r \in R'_t} \sum_{l \in L'_t} c_{rl} x_{rl} \\
\text{subject to } & \sum_{l \in L'_t} x_{rl} \leq 1 \text{ for each } r \in R'_t; \\
& \sum_{r \in R'_t} x_{rl} \leq 1 \text{ for each } l \in L'_t; \\
& x_{rl} \in \{0,1\} \text{ for all } r \in R'_t, l \in L'_t.
\end{aligned}$$

Then we define the value (gradient) of a resource to be

$$v_{rt} = \begin{cases} C(S_t) - C(S_t - r) & \text{if } r \in S_t; \\ C(S_t + r) - C(S_t) & \text{if } r \notin S_t. \end{cases}$$

Similarly, we define the value (gradient) of a task to be

$$v_{lt} = \begin{cases} C(S_t) - C(S_t - l) & \text{if } l \in S_t; \\ C(S_t + l) - C(S_t) & \text{if } l \notin S_t. \end{cases}$$

(For notational simplicity we write $S_t - r$ rather than $S_t - \{r\}$ and $S_t - l$ instead of $S_t - \{l\}$. Similarly, we write $S_t - rl$ rather than $S_t - \{r, l\}$. In addition, our assumption that $R$ and $L$ are disjoint implies that $v_{rt}$ and $v_{lt}$ remain well-defined.)

The resource and task gradients are closely related to the dual variables. In fact, the common interpretation of a dual variable for a resource or task is that it represents the marginal value of that resource or task. However, this interpretation fails in the presence of degeneracy. Instead, if $u_r$ is the dual variable associated with resource $r$ in a particular optimal solution for a network $S$, $C(S+r)-C(S) \le u_r \le C(S)-C(S-r)$. (See, for example, Aucamp and Steinberg [13].) Moreover, solutions to the assignment problem exhibit a large degree of degeneracy. Thus using the dual variables from time $t+1$ may not accurately capture the true value of having resources or tasks available at time $t+1$. Our resource and task gradients avoid the problems with degeneracy, as we now prove.

## 3. Proofs

First, we need a theorem of Shapley's [14].

**Theorem 1** (Shapley)**.** *Given a network $S$, $\Big(C(S+r)-C(S)\Big)+\Big(C(S+l)-C(S)\Big) \le C(S+rl)-C(S)$.*

In Shapley's terminology, resources and tasks are *complements*, i.e., the value to the network of adding a resource and task together is greater than the sum of the values of adding the resource and task separately.

Now we prove our main result that the resource and task gradients are optimal discounts.

**Theorem 2.** *Let $\mathbf{x}^*$ denote the optimal solution to a given offline assignment problem. Let $\mathbf{x}$ denote the solution to the corresponding online assignment problem found by the greedy method when the contributions $c_{rl}$ are discounted by $v_{r,t+1}+v_{l,t+1}$. Then, if $\mathbf{x}^*$ is unique, then $\mathbf{x}=\mathbf{x}^*$.*

*Proof.* Assuming that $\mathbf{x}$ makes the same assignments as does $\mathbf{x}^*$ for times $t'<t$, and if task $l$ becomes available at time $t$, then we need to show

1. If $x^*_{rlt}=1$ then $x_{rlt}=1$.
2. If $x^*_{rlt}=0$ for all $r$, then $x_{rlt}=0$ for all $r$ available at time $t$.

We accomplish this by proving the following claims.

*Claim 1.* If $x^*_{rlt}=1$, then $v_{r,t+1}+v_{l,t+1}<c_{rl}$.

*Claim 2.* If $x^*_{rlt}=1$, then, for all $r'$ available at time $t$ with $r'\ne r$, $v_{r',t+1}+v_{l,t+1}\ge c_{r'l}$.

*Claim 3.* If $x^*_{rlt}=0$ for all $r$, then, for all $r$ available at time $t$, $v_{r,t+1}+v_{l,t+1}>c_{rl}$.

Claims 1 and 2 together establish that if $x^*_{rlt}=1$, then $r$ is the only resource available at time $t$ for which $c_{rl}-v_{r,t+1}-v_{l,t+1}>0$. Thus $x_{rlt}=1$ as well. Claim 3 shows that if $x^*_{rlt}=0$ for all $r$ available at time $t$, then $c_{rl}-v_{r,t+1}-v_{l,t+1}<0$ for all $r$ available at time $t$. Thus $x_{rlt}=0$ for all $r$ available at time $t$ as well.

*Proof of Claim 1.* Let $l'$ and $r'$ denote, respectively, the task to which $r$ and the resource to which $l$ are assigned in the optimal solutions to the networks $S_{t+1}+r$ and $S_{t+1}+l$, respectively. (It may be that $r$ and $l$ are unassigned in these networks, but the argument still holds.) Then $C(S_{t+1}+r)=C(S_{t+1}-l')+c_{rl'}$, and $C(S_{t+1}+l)=C(S_{t+1}-r')+c_{r'l}$. By uniqueness, $C(S_{t+1}-r'l')+c_{rl'}+c_{r'l}<C(S_t)=C(S_{t+1})+c_{rl}$. By Theorem 1, $C(S_{t+1}-r')+C(S_{t+1}-l')\le C(S_{t+1})+C(S_{t+1}-r'l')$. Putting all this together, we have

$$
\begin{aligned}
&C(S_{t+1}+r)+C(S_{t+1}+l)\\
&=C(S_{t+1}-l')+c_{rl'}+C(S_{t+1}-r')+c_{r'l}\\
&\le C(S_{t+1})+C(S_{t+1}-r'l')+c_{rl'}+c_{r'l}<2C(S_{t+1})+c_{rl}\\
&\Rightarrow C(S_{t+1}+r)-C(S_{t+1})+C(S_{t+1}+l)-C(S_{t+1})<c_{rl}\\
&\Rightarrow v_{r,t+1}+v_{l,t+1}<c_{rl}.
\end{aligned}
$$

*Proof of Claim 2.* Since $x_{r'l}=1$ is a feasible assignment for $S_{t+1}+l$, $C(S_{t+1}+l)\ge C(S_{t+1}-r')+c_{r'l}$. Then $v_{r',t+1}+v_{l,t+1}=C(S_{t+1})-C(S_{t+1}-r')+C(S_{t+1}+l)-C(S_{t+1})\ge -C(S_{t+1}-r')+C(S_{t+1}-r')+c_{r'l}=c_{r'l}$.

*Proof of Claim 3.* Since $l$ was not assigned at time $t$, any resource $r$ available at time $t$ is also available at time $t+1$. In addition, $x_{rl}=1$ is a non-optimal feasible assignment for $S_{t+1}$. Since the optimal solution is unique, $C(S_{t+1})>C(S_{t+1}-rl)+c_{rl}$. Then $v_{r,t+1}+v_{l,t+1}=C(S_{t+1})-C(S_{t+1}-r)+C(S_{t+1})-C(S_{t+1}-l)>C(S_{t+1})-C(S_{t+1}-r)+C(S_{t+1}-rl)+c_{rl}-C(S_{t+1}-l)\ge c_{rl}$, where the last step follows from Theorem 1. □

From Theorem 2 it is easy to show that the resource gradients alone also constitute optimal discounts.

**Corollary 1.** *Let $\mathbf{x}^*$ denote the optimal solution to a given offline assignment problem. Let $\mathbf{x}$ denote the solution to the corresponding online assignment problem found by the greedy method when the contributions $c_{rl}$ are discounted by $v_{r,t+1}$. Then, if $\mathbf{x}^*$ is unique, then $\mathbf{x}=\mathbf{x}^*$.*

*Proof.* Suppose task $l$ becomes available at time $t$.

Claims 1 and 2 from Theorem 2 imply that if $x^*_{rlt}=1$, then $c_{rl}-v_{r,t+1}>c_{r'l}-v_{r',t+1}$ for any $r'$ available at time $t$ with $r'\ne r$. So assigning $r$ to $l$ is better than assigning any other resource to $l$. They also imply that $c_{rl}-v_{r,t+1}>0$, which means that assigning $r$ to $l$ is better than making no assignment. Thus if $x^*_{rlt}=1$, then $x_{rlt}=1$ as well.

If $x^*_{rlt}=0$ for all $r$, then $C(S_t)-C(S_t-l)=0$. But no assignments at time $t$ means that $S_t=S_{t+1}$, and so $v_{l,t+1}=C(S_{t+1})-C(S_{t+1}-l)=0$ as well. Then Claim 3 from Theorem 2 implies that for all $r$ available at time

$t$, $v_{r,t+1} > c_{rl}$. Thus if $x^*_{rlt} = 0$ for all $r$ then the best solution at time $t$ is not to assign $l$ to any resource, which yields $x_{rlt} = 0$ for all $r$ available at time $t$. □

## 4. Final comments

Our earlier paper [10] showed only that the resource gradients give optimal discounts. We were unable to adapt our proof there to the case of the resource and task gradients. In fact, we do not even mention the resource and task gradients in that paper; the experimental results showing that the resource and task gradients give optimal discounts are instead in Spivey's doctoral thesis [15]. We see here, though, that the proofs are much easier if we prove the result first for the resource and task gradients; from there the proof for the resource gradients alone is a fairly straightforward corollary. Thinking of the resource and task gradients as close approximations of the dual variables turns out to be an important insight.

Our earlier paper also proves that discounting with resource gradients gives the optimal offline solution for problem instances in which tasks are allowed to remain in the network if not assigned but for which the contribution for assigning resource $r$ to task $l$ is a strictly decreasing function of $t$. Our results here easily extend to this case as well.

Some questions for future research include the following.

- Theorem 2 and Corollary 1 assume that the optimal offline solution is unique. In the absence of a unique optimal offline solution, the proof of Theorem 2 implies that the discounts derived from an optimal offline solution $\mathbf{x}^*$ might produce a situation in which $c_{rl} - d_{rlt} = c_{r'l} - d_{r'lt} = 0$, even though $x^*_{rlt} = 1$ and $x^*_{r'lt} = 0$. This can happen, for instance, if $c_{rl} = c_{r'l}$, and $r'$ is not assigned to any task under $\mathbf{x}^*$, while another optimal solution $\mathbf{x}'$ has exactly the same assignments as does $\mathbf{x}^*$ except that $x'_{r'lt} = 1$ and $r$ is unassigned to any task. In the case of such a tie for the choice of time $t$ assignment, the algorithm could assign $r$ or $r'$ or even no resources to $l$ at time $t$. Does the assignment of $r'$ to $l$ in such a scenario always lead to a different optimal solution, or is it possible that a "wrong" choice when faced with a tie could result in a suboptimal solution? (We could require the algorithm to rule out the no-assignment option, which would definitely lead to a suboptimal solution.) More generally, can we describe the behavior of our discounted greedy algorithms in the presence of multiple optimal offline solutions?

- Can we characterize the class of discount functions for which the greedy method is exact?

- Can these results be extended to generalizations or variations of the standard assignment problem, such as the transportation problem and the bottleneck assignment problem?

## References


[1] R. M. Karp, U. V. Vazirani, V. V. Vazirani, An optimal algorithm for on-line bipartite matching, in: STOC '90: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing, ACM Press, 1990, pp. 352–358.

[2] B. Kalyanasundaram, K. Pruhs, Online weighted matching, Journal of Algorithms 14 (3) (1993) 478–488.

[3] B. Birnbaum, C. Mathieu, On-line bipartite matching made simple, ACM SIGACT News 39 (2008) 80–87.

[4] J. Feldman, A. Mehta, V. Mirrokni, S. Muthukrishnan, Online stochastic matching: beating $1 - 1/e$, in: Symposium on the Foundations of Computer Science (FOCS), 2009.

[5] E. Vee, S. Vassilvitskii, J. Shanmugasundaram, Optimal online assignment with forecasts, in: Proceedings of the 11th ACM Conference on Electronic Commerce, 2010.

[6] R. Burkard, M. Dell'Amico, S. Martello, Assignment Problems, Society for Industrial and Applied Mathematics, Philadelphia, 2009.

[7] D. J. Aldous, The $\zeta(2)$ limit in the random assignment problem, Random Structures and Algorithms 18 (2001) 381–418.

[8] A. J. Hoffman, On simple linear programming problems, in: Proceedings of Symposia in Pure Mathematics, Vol. 7, American Mathematical Society, 1963, pp. 317–327.

[9] M. Z. Spivey, W. B. Powell, The dynamic assignment problem, Transportation Science 38 (4) (2004) 399–419.

[10] M. Z. Spivey, W. B. Powell, Some fixed-point results for the dynamic assignment problem, Annals of Operations Research 124 (2003) 15–33.

[11] A. George, W. B. Powell, S. R. Kulkarni, Value function approximation using multiple aggregation for multiattribute resource management, Journal of Machine Learning Research 9 (2008) 2079–2111.

[12] W. B. Powell, Approximate Dynamic Programming, John Wiley & Sons, 2007.

[13] D. C. Aucamp, D. I. Steinberg, The computation of shadow prices in linear programming, The Journal of the Operational Research Society 33 (1982) 557–565.

[14] L. S. Shapley, Complements and substitutes in the optimal assignment problem, Naval Research Logistics Quarterly 9 (1) (1962) 45–48.

[15] M. Z. Spivey, The dynamic assignment problem, Ph.D. thesis, Princeton University (2001).