A Reinforcement Learning Approach to Constrained Resource Allocation Problems

C. Vic Hu
Department of Electrical & Computer Engineering
University of Texas at Austin

***Abstract*—Resource allocation has been extensively studied in various fields such as wireless communication, intelligent traffic routing, industrial management, parallel architectures and distributed systems. Although there have been many efficient and powerful algorithms proposed in each domains, very few of them is able to address more generalized resource allocation problems on a higher level.**

This paper proposes a generalized framework and makes three main contributions to solving constrained resource allocation problem using a modified reinforcement learning algorithm. First, we designed a general architecture, Constrained Resource Allocation Framework (CRAF), for the ease of generalized problem mapping and learning. Second, we defined four properties and three measurements to both quantitatively and qualitatively analyze how well an algorithm performs in CRAF. Finally, we developed three benchmark experiments to demonstrate how a reinforcement algorithm can successfully solve very different resource allocating problems with CRAF.

I. Introduction

Resource allocation is an old and widely-solved problem in many fields, including electrical engineering, computer science, economics, management science and many more. It typically involves a fixed number of resource units to be distributed to a set of tasks over a period of time, such as landing aircraft scheduling, wireless communication routing, social security welfare and shared resources in parallel computer architectures. The problems of resource allocating are ubiquitous, but they all essentially boil down to one simple notion–based on a set of criteria, how many resource units should be allocated to which task first, and for how long.

To name a few remarkable research examples to solve this type of problems, Dresner and Stone proposed s reservation system for autonomous intersection management [1] to allocate right of road for crossing traffic, Perkins and Royer presented a novel routing algorithm for ad-hoc on-demand mobile nodes management [2], and Foster et al. came up with a reservation and allocation architecture for heterogeneous resource management in computer network. While they all addressed an elegant solution to a specific domain, most of the techniques cannot be easily transferred to similar decision problems in other domains. In this paper, we focus on addressing exactly this issue and forming a general resource allocation framework that is suitable to be solved by a reinforcement learning method.

The first contribution of this paper is to form the Constrained Resource Allocation Framework (CRAF), in which we designed a generalized architecture to capture most of the resource allocation problems. In addition to the conventional first-come, first-served basis, we introduced the notion of constraints and queue propagation to reflect a more realistic setting and to relax more complicated systems into a single-frontier problem.

Secondly, we defined a collection of analysis criteria and evaluation methods to both qualitatively and quantitatively study how a reinforcement learning algorithm perform on our framework. Lastly, we proposed three benchmark problems to empirically demonstrate how CRAF applies to different resource allocation problems consistently and effectively.

II. Background

Unlike the Markov Decision Processes (MDPs) that most of the reinforcement learning algorithms are designed to solve, the notion of state representation, transition functions and reward functions in resource allocation problems such as aircraft landing scheduling can be very complex and challenging to define. Furthermore, the state space representing the entire global snapshot could be too large to be useful for effective learning.

Instead of trying to model a resource allocation problem as MDP, we found it more intuitive to formalize it as a multi-armed bandit problem, which has been extensively researched in the field of reinforcement learning [6]. Assuming that we

can reasonably classify each incoming task candidate to one of the K prototypes, from which we use the approximated reward functions to determine which candidate receives the resource unit in each episode. Before we formalize our definitions and notations of the framework, let us go through some of the existing K-armed algorithms and see how they can be useful to the resource allocation problem.

A. The K-armed Bandit Problem

The problem is formalized by a fixed number of slot gambling machines, each defined by a random variable $X_{i,n}$, for $1 \leq i \leq K$ and $n \geq 1$. A sequential N plays of machine i give rewards of $X_{i,1}, X_{i,2}, \ldots, X_{i,N}$, which are all independent of each other and identically distributed. Based on the sequence of playing history and obtained rewards, one can form an allocation algorithm to pick the next machine. To evaluate the performance of such algorithms, one criterion, the $regret$, is defined as

$$regret = \mu^* n - \mu_j \sum_{j=1}^{K} E[T_j(n)]$$

where $\mu^* \equiv \max_{1 \leq i \leq K} \mu_i$, $T_j(n)$ is the number of times machine j has been played during the n plays. Therefore, $regret$ is essentially the expected loss function (opportunity cost) of the played allocation strategy.

III. The Constrained Resource Allocation Framework

Formally, the Constrained Resource Allocation Framework (CRAF) consists of a sequential task instances, and a centralized learning agent who distributes resource units to the task instances and receives rewards according to the decision they make. The task instances can come in one or more channels $\mathbf{x}_{c,i}$, $1 \leq c \leq C$, $1 \leq i \leq capacity(c)$, where i denotes the i-th instance and c denotes the c-th channel. To transform a resource allocation problem into a learnable framework, the formalization process under CRAF can be broke down into three phases:

1) Housekeeping Phase
   - Constraint
   - Objective
2) Tuning Phase
   - Priority Score
   - Queue Propagation
   - Failure Capture
3) Prototyping Phase
   - Prototype Classification
   - Reward Function

IV. Experimental Results

1. 2-way intersection (simplest form) 2. 4-way intersection (special corner cases) 3. n-children candy distribution (multi-channel)

V. Discussion and Related Work

VI. Conclusion

VII. Future Work

VIII. Acknowledgements

References

[1] Dresner, K. and Stone, P. A Multiagent Approach to Autonomous Intersection Management. *Journal of Artificial Intelligence Research*, 31:591-656, March 2008.

[2] Perkins, C. E., and Royer, E. M. (1999, February). Ad-hoc on-demand distance vector routing. In *Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA'99. Second IEEE Workshop on (pp. 90-100).* IEEE.

[3] Foster, I., Kesselman, C., Lee, C., Lindell, B., Nahrstedt, K., and Roy, A. (1999). A distributed resource management architecture that supports advance reservations and co-allocation. In *Quality of Service, 1999. IWQoS'99. 1999 Seventh International Workshop on (pp. 27-36).* IEEE.

[4] Baruah, S. K., Cohen, N. K., Plaxton, C. G., and Varvel, D. A. (1996). Proportionate progress: A notion of fairness in resource allocation. *Algorithmica*, 15(6), 600-625.

[5] Thomas, D. (1990). Intra-household resource allocation: An inferential approach. *Journal of human resources*, 635-664.

[6] Auer, P., Cesa-Bianchi, N. and Fischer, P. Finite-time Analysis of the Multiarmed Bandit Problem. In *Proc. of 15th International Conference on Machine Learning, pages 100-108*. Morgan Kaufmann, 1998.

[7] Diuk, C., Li, L. and Leffler, B. R. The Adaptive k-Meteorologists Problem and Its Application to Structure Learning and Feature Selection in Reinforcement Learning. In *Proceedings of the 26th International Conference of Machine Learning*. Montreal, Canada, 2009.