Multi-agent Resource Allocation with Pre-emption for Dynamic Future Arrivals

Graham Pinhey
gpinhey@uwaterloo.ca

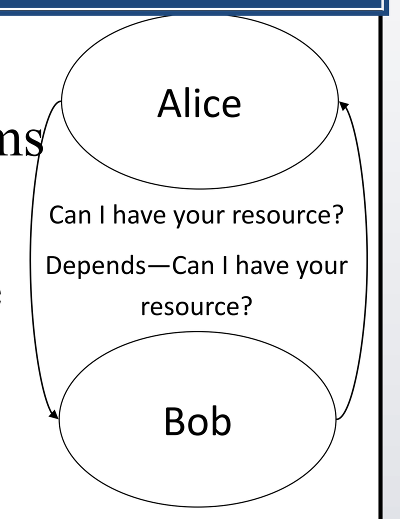
University of Waterloo

Supervisor: Dr. Robin Cohen
rcohen@uwaterloo.ca

1. Introduction

- Multi-agent: Every user has an AI Agent acting on their behalf.
- Resource allocation: Limited supply of something (e.g. time) shared among these agents. Co-operative setting, goal is to maximize global utility: Satisfy user demands, some users objectively more important.
- Pre-emption: Reallocate resources when greater needs arise (largely unsolved). Allows system to react to changes in the environment.
- Challenge: Anticipate future needs to avoid bad allocations.
- Approach: Opportunity cost, learn expected demand, keep a reserve.

2. Background



- Starting point: Doucette (2012) model [1], reason with Transfer of Control strategies to avoid problems with cycles in pre-emption—Closed-form estimate of backup plan based on chance of getting resource
- Learning: Congestion (can you get backup?) and Churn (how far ahead should you plan?)
- Motivating application: Allocating doctors to patients in hospitals
- Proxy Agents: AI filter to avoid bothering humans unless necessary

3. Example

- Resources: Doctors. Tasks: Patients, arriving randomly over time.
- Say you have two expert surgical teams and one trainee team. If you have two low-severity patients now, who works on them? The experts will finish faster and better than trainees, but what if a high-severity patient shows up in 15 minutes that the trainees can't handle?
- With pre-emption, you abandon work on one patient to handle the new arrival. This loses progress and adds task-switching cost.
- If new arrivals are expected, you should consider holding teams back.

4. Model Improvements

- Dynamic changes over time: Previous currency-based work [2] was very sensitive to order of arrival, injecting more currency if high-priority tasks arrive late. Current system only values fairness.
- Backup plans: The ability of an agent to secure backup resources is considered when taking a resource.
- Agents representing resources filter incoming requests to only pass on the most effective request—speeds up finding best allocation

5. Opportunity Cost of Action

- Expected opportunity cost of locking resources from time t_0 to t_1:

$$opportunity\_cost(s_0, t_0, t_1) \leq \sum_{t=t_0+1}^{t_1} \left[\sum_{n=1}^{\infty} [P_a(t,n) * n] * \sum_{s=s_0}^{s_{max}} [P_s(t,s) * \Delta EU(t, s, s_0, t_0, t_1)] \right]$$

$$\Delta EU(t, s, s_0, t_0, t_1) = \max \{0, [EU(t+c, t_1+c, \theta_s, 0) + EU(t+c+t_1-t_0, t_1, \theta_{s_0}, t_1-t_0)]\}$$

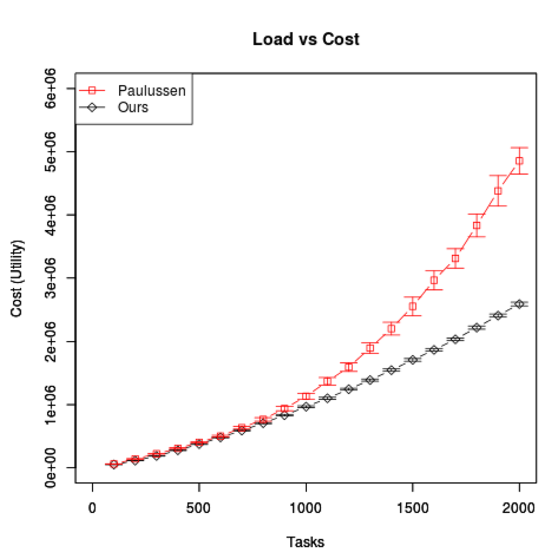
- $P_a(t,n)$: Probability that n tasks will arrive at time t
- $P_s(t,s)$: Probability that tasks arriving at time t have type θ_s
- $EU(t_{new},t_{old},\theta_s,t_{wait})$: Change in expected utility from starting a task of type θ_s waiting for t_{wait} at time t_{new} instead of time t_{old}
- This focuses on atomic (no pre-emption) case for simplicity

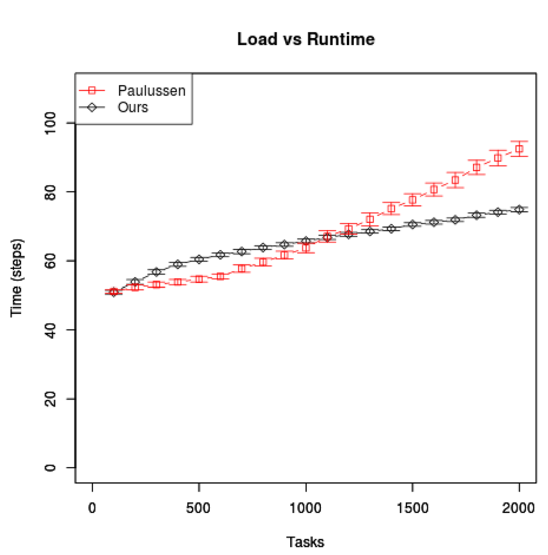
6. Predicting Future Arrivals

- To get P_a and P_s, introduce a new learning agent: Triage Agent, tracking incoming arrivals to the system and predicting future arrivals.
- Learning algorithms usually focus on eventual correctness, finding an underlying distribution in the equilibrium case.
- Here, interesting cases are at non-equilibrium—medical mass casualty incidents, oversubscribed schedules in real-time scheduling…
- Exploratory work will examine value of perfect distribution knowledge before attempting to learn short-term distributions.

7. Results

- Simulations run with 100 resources, for 100 trials/point.
- X-axis: Number of tasks, arriving uniformly distributed over 50 steps.





- Y-axis: Total cost or time required to finish all tasks. Low is better.

8. Analysis

- In low-load cases (<1000 tasks) both algorithms have similar cost, as the system has more than enough capacity for everyone.
- When load is high, older algorithms grow in cost faster than the new algorithm as number of patients increases.
- The new algorithms have less variance. Agent request filtering makes allocation more predictable.
- Highly congested systems are difficult to handle with current techniques, there is likely further room for improvement.

9. Conclusion and Next Steps

- Advances field of multi-agent resource allocation
- Designing with changing environment in mind provides benefits
- Future work: Run more simulations to determine value of predicting future arrivals

Reference

[1] J. Doucette, "An ex-ante rational distributed resource allocation system using transfer of control strategies for preemption with applications to emergency medicine," Master's thesis, University of Waterloo, 2012.

[2] T.O. Paulussen, A. Zöller, A. Heinzl, A. Pokahr, L. Braubach, and W. Lamersdorf, "Dynamic patient scheduling in hospitals," *Agent Technology in Business Applications*, 2004.

WATERLOO
CHERITON SCHOOL OF COMPUTER SCIENCE

hSITE
Healthcare Support through Information Technology Enhancements