Towards Addressing Dynamic Multi-agent Task Allocation in Law Enforcement

JAAMAS Track

Itshak Tkach
London University
London, UK
i.tkach@gold.ac.uk

Sofia Amador Nelke
Holon Institute of Technology
Holon, Israel
sofiaa@hit.ac.il

ABSTRACT

To deal with the underlying heterogeneous *law enforcement problem* (LEP^H), one needs to allocate police officers to dynamic tasks whose locations, arrival times, and importance levels are unknown a priory. Addressing this challenge and inspired by real police logs, this research aims to solve the LEP^H problem by using and comparing three methods: Fisher market-based FMC_TA^{H+}, swarm intelligence HDBA, and Simulated Annealing SA algorithms. The three methods were compared in this study for the performance measures that are commonly used by law enforcement authorities. The results indicate an advantage for FMC_TA^{H+} both in total utility and in the average arrival time to tasks. Also, compared respectively to *HDBA* and *SA*, FMC_TA^{H+} leads to 34% and 32% higher team utility in the highest shift workload.

KEYWORDS

Multi-agent systems; artificial intelligence; law enforcement problem

ACM Reference Format:
Itshak Tkach and Sofia Amador Nelke. 2022. Towards Addressing Dynamic Multi-agent Task Allocation in Law Enforcement : JAAMAS Track. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 2 pages.

1 INTRODUCTION

LEP^H is an assignment problem where a team of cooperative heterogeneous agents, each with a possibly different set of skills and a different utility value, has a common goal (e.g., police officers that specialize in defusing bombs and that have dogs capable of detecting drugs). The law enforcement tasks associated with LEP^H require specific combinations of agent skills. The goal of this paper is to meet the challenge of solving the realistic LEP^H problem based on real police logs by applying and comparing Fisher market-clearing task allocation (FMC_TA^{H+}) algorithm[1], heterogeneous distributed bees algorithm (*HDBA*) in swarm intelligence[2], and an algorithm based on the simulated annealing framework, which is a general-purpose framework for optimization (SA). In realistic applications such as LEP^H, avoiding dependence on a central dispatch for coordinating task allocation is preferred. In a disaster scenario, for instance, communication lines to a central location may break down and a single-point-of-failure is preferably avoided, especially in malicious scenarios (e.g. a terror attack). Thus, an important objective of this paper when designing a task allocation algorithm for these types of applications is that it can be used in distributed scenarios as well as in centralized settings. In the setting presented in this paper, the information regarding crime incidents is shared among agents within a specific radius.

2 PROBLEM DEFINITION

The allocation of multiple agents to unknown tasks arriving at unknown times and locations considers both the tasks to be performed and the skills an agent possesses. If more than one skill is required in performing the task, several agents must cooperate sequentially. If different skills are required, agents can perform the task concurrently. Formally, the allocation of tasks to agents in LEP^H is a matrix X where entry x_{iks} is the fraction of task Ψ_i assigned to agent k, utilizing the skill s. Each agent's schedule must include, both the task being performed and the start and end times, with the utilized skill. Thus, each member of the schedule σ_k of agent k includes (Ψ_i, s, t, t') specifying, respectively, the task, the utilized skill, and the start and end times for applying skill s on task Ψ_i by agent k. Besides, inter-skill constraints that require concurrence between police officers with different skills are employed. The performance of task Ψ_i depends on the number of agents that work simultaneously on that task. An agent gets utility for performing a task by using a skill s. Thus a capability function is defined for a vector $\vec{q} \in N$ that specifies the number of agents with each skill working concurrently on a task, i.e., the l'th entry in $\vec{q}$ represents the number of agents with skill s_l working on the task concurrently. Each agent can be counted only once, i.e., they cannot utilize multiple skills simultaneously. The result of the function $Cap\,(\Psi_i, \vec{q})$ is a vector $\vec{g}$ specifying for each skill the utility derived by an agent performing the task, taking into consideration the number of agents using this skill. Let $d\,(\Psi_i, \vec{q})$ be the time duration that $\vec{q}$ represents the set of agents working simultaneously on task Ψ_i Thus, $\frac{d(\Psi_i,\vec{q})}{w(\Psi_i)}$ the relative portion of time that the set of agents specified by $\vec{q}$ are working on Ψ_i where $w\,(\Psi_i)$ is the workload must be performed to complete the task Ψ_i. Denote by $\vec{Q}$ the set of all possible vectors $\vec{q}$. The utility derived by the agents for completing Ψ_i is:

$$U' = \sum_{\vec{q} \in \vec{Q}} \frac{d\,(\Psi_i, \vec{q})}{w\,(\Psi_i)} \sum_{l=1}^{s} q[l]g[l] \tag{1}$$

where $q[l]$ and $g[l]$ are the l'th entry in vectors $\vec{q}$ and $\vec{g}$ respectively. The utility derived for completing task Ψ_i starting at time t_{Ψ_i} depends on the capability of the agents performing the task and the

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

soft deadline function $\delta(\Psi_i, t) = \beta^{\gamma(t_{\psi_i} - t_i^{arrival})}$. The discounted utility for performing task ψ_i with arrival time at a time $t_i^{arrival}$ and which is initially handled at time t_{ψ_i} is:

$$U'_{\Psi_i} = \beta^{\gamma(t_{\Psi_i} - t_i^{arrival})} \sum_{\vec{q} \in \vec{Q}} \frac{d(\Psi_i, \vec{q})}{w(\Psi_i)} \sum_{l=1}^{s} q[l]g[l] \quad (2)$$

where $\beta \in (0,1)$ and $\gamma \geq 0$ are constants.

When a new task arrives, the current task (if any) being performed by agent k is denoted CT_k and the current skill that is used by agent k for CT_k is denoted CS_k. Agents can interrupt the performance of their current task. The penalty for task interruption is $\pi(\Psi_i, \Delta w_i^{CS_k})$, which depends on the task Ψ_i and the amount of work $\Delta w_i^{CS_k}$ for skill CS_k completed when the task is interrupted. The adjusted penalty for task Ψ_i decreases exponentially with $\Delta w_i^{CS_k}$ to a minimum value:

$$\pi(\Psi_i, \Delta w_i^{CS_k}) = \max\{I(\Psi_i) c^{w_i^{CS_k} - \Delta w_i^{CS_k}}, \phi \cdot I(\Psi_i)\} \quad (3)$$

where $c \in (0,1)$ and $\phi > 0$ are constants and $I(\Psi_i)$ is the importance of task Ψ_i. The total utility derived for performing Ψ_i is thus:

$$U(\Psi_i) = U'(\Psi_i) - \sum_{a_k : \Psi_1^k \neq CT_k} \pi(CT_k, \Delta w_i^{CS_k}) \quad (4)$$

The experimental design resembles a realistic LEP^H in our hometown. The city was represented by a rectangular region of the Euclidean plane of size 10×10 kilometers, divided into 25 neighborhoods of size 2×2 kilometers, each with a patrol task. The setup includes 8-hour shifts (as in the real police department), with 25 agents patrolling (one in each neighborhood) at the beginning of each shift. The number of tasks arriving (i.e., the load) in a shift varied between 56, 111, 167, 222, and 278. Tasks arrived at a fixed rate and were uniformly distributed in random locations in the city. The tasks are used as input and the utilities as the output of the algorithms. Upon the arrival of each task, algorithms are executed to compute the allocation strategy to maximize the total utility.

3 RESULTS

Figure 1 graphically represents and compares the results of the simulations for the three algorithms as a function of shift load. It illustrates as expected that, at the lower shift loads of 55, 111, and 167, the team utility increases with the load for all three algorithms. However, at the higher loads of 222 and 278, the team utility continues to increase for FMC_TA^{H+} but decreases for *HDBA* and *SA*. This indicates that agents started to perform less efficiently and have decreased the overall profit from handling tasks. For the high loads of 222 and 278, the difference in team utility between FMC_TA^{H+} and two other algorithms is significant ($p\_value < 0.05$), with average team utility values for FMC_TA^{H+}, *HDBA* and *SA* being 198713.6, 129849.1, and 134892.9 respectively.

4 CONCLUSION

This research addressed the challenge of solving the LEP^H problem inspired by real police logs by allocating heterogeneous police

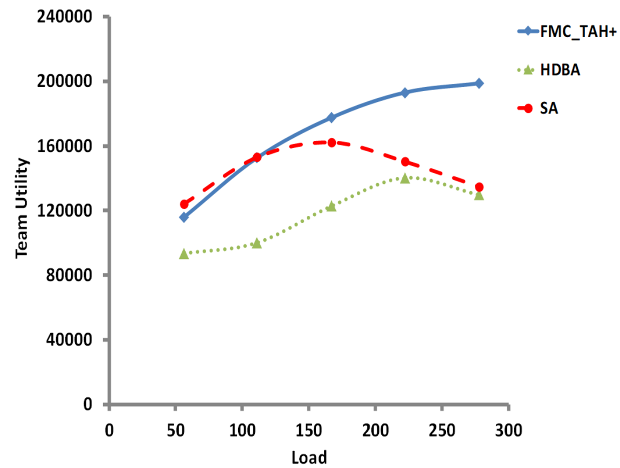

Figure 1: Team utility as a function of shift load.

officers to dynamic tasks whose locations, arrival times, and importance levels are unknown a priory. Realistic measurements were obtained by consulting police officers about specific evaluation metrics that are of interest to them. The common ground for this problem is that the agents are cooperative, and they have a common goal - to maximize the team utility, which is mostly hard to quantify. The described approach to handle scenarios with heterogeneous agents who possess different skills, by including a product for each skill required for each task, as shown in the present study to be effective in allocating dynamic tasks to heterogeneous police agents using simulation analyses. All algorithms were shown to solve the dynamic task allocation problem. The FMC_TA^{H+} algorithm has generated more cooperation among agents and resulted in better performance in terms of team utility and execution delay than both HDBA and SA, with statistically significant 34% and 32% higher team utility in the highest shift load compared to *HDBA* and *SA* respectively. This result indicates that by using FMC_TA^{H+}, police officers were able to deal with crime incidents faster and more efficiently before they got outdated. The algorithm's performance results in allocations that share important tasks, enabling effective cooperation that leads to higher quality task execution while minimizing delays. This approach could be applied in many applications similar to LEP^H, such as fire fighting, surveillance, search and rescue, military and homeland security, among others.

REFERENCES

[1] Sofia Amador Nelke and Roie Zivan. 2017. Incentivizing Cooperation between Heterogeneous Agents in Dynamic Task Allocation.. In *AAMAS*. 1082–1090.
[2] Itshak Tkach and Yael Edan. 2020. Extended examples of single-layer multi-sensor systems. In *Distributed Heterogeneous Multi Sensor Task Allocation Systems*. Springer, 49–79.