Assessing Routing Decisions of Search and Rescue Teams in Service of an Artificial Social Intelligence Agent

Volkan Ustun[1] [a], Ronit Jorvekar[2], Nikolos Gurney[1] [b], David Pynadath[1,2] [c] and Yunzhe Wang[3] [d]

[1] *Institute for Creative Technologies, University of Southern California, Playa Vista, CA, U.S.A.*

[2] *Department of Computer Science, University of Southern California, Los Angeles, CA, U.S.A.*

[3] *Department of Computer Science, Columbia University, New York, NY, U.S.A.*

Keywords: Artificial Social Intelligence, Graph Neural Networks, Route Planning, Urban Search and Rescue.

Abstract: In the context of Urban Search and Rescue (USAR) missions, efficient routing performance is of paramount importance for the success of a USAR team. Artificial Social Intelligence (ASI) agents could play a crucial role in guiding and interacting with these teams, and an analysis of the routing choices made by USAR teams can offer valuable insights into their overall performance and provide guidance for interventions by ASI agents. This study capitalizes on recent advancements in Graph Neural Networks, transformers, and attention models to harness their capabilities as neural heuristics for rapidly generating near-optimal routes in routing challenges. Specifically, we propose a real-time decision framework to scrutinize and evaluate routing decisions executed by participants during the DARPA ASIST Minecraft USAR Task. This assessment involves comparing the routing decisions made by participants and routes concurrently generated and recommended by neural heuristics employing Graph Neural Networks with attention mechanisms. Furthermore, our investigation delves into the potential of routing decision assessments as informative indicators for an ASI agent, aiding in identifying scenarios necessitating intervention. This research contributes to using quantitative metrics, such as routing efficiency, as meaningful signals for ASI agents to monitor the performance of USAR teams through integrating state-of-the-art AI techniques. Ultimately, this integration could enhance the efficiency and effectiveness of an ASI in guiding search and rescue operations.

1 INTRODUCTION

Autonomous agents have the potential to enhance teamwork by automatically assessing and providing assistance during task performance (Sukthankar et al., 2007; Webber et al., 2019). Furthering research in this domain is the objective of DARPA's ASIST program, which employs a simulated urban search and rescue (USAR) task using a Minecraft environment to evaluate autonomous agents (Freeman et al., 2023). This task environment is utilized to run experiments with distributed teams of three participants aided by an Artificial Social Intelligence (ASI) agent acting as an advisor. The success and safety of the USAR team are heavily reliant on routing decisions, necessitating the tracking of participant and team routing decisions by a capable USAR ASI agent.

A USAR ASI agent can use routing in multiple ways. For instance, it can aid with real-time routing decisions of search and rescue teams by utilizing the outputs of a route generator resembling a navigation app that creates routes and recommends alternatives based on present data such as road conditions, traffic congestion, accidents, and more. However, humans do not always strictly adhere to the recommendations of a navigation app because they consider their own private information, including preferences and their physiological and emotional states. To illustrate, a driver with a car full of hungry kids may choose to stop and eat dinner early after the app recommends a re-route rather than following the recommendation. Similar to how the app performs real-time path planning and offers choices to the user based on current data, a USAR ASI agent can ingest routing suggestions and merge them with other mission-related information and the affective state of the USAR team to provide more effective guidance.

[a] https://orcid.org/0000-0002-7090-4086
[b] https://orcid.org/0000-0003-3479-2037
[c] https://orcid.org/0000-0003-2452-4733
[d] https://orcid.org/0000-0003-3862-1163

Ustun, V., Jorvekar, R., Gurney, N., Pynadath, D. and Wang, Y.
Assessing Routing Decisions of Search and Rescue Teams in Service of an Artificial Social Intelligence Agent.
DOI: 10.5220/0012388100003636

In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 1*, pages 313-320
ISBN: 978-989-758-680-4; ISSN: 2184-433X

Another example of how an ASI agent can leverage routing is to monitor the team's routing decisions to assess the team's overall state. For instance, a decline in the quality of routing decisions may indicate potential issues within the search and rescue team or suggest that the team is not fully leveraging available information. The ASI agent can use these observations and inferences to provide improved suggestions and interventions, ultimately enhancing the team's performance.

Generating good routing options in real-time is critical for both use cases mentioned above, as they require considering the current state of the USAR mission. However, this task is challenging because ASIST's USAR routing is similar to the family of NP-Hard Traveling Salesman Problems (TSPs). As a result, heuristics must be employed to find candidate solutions, as is the case with many practical applications (Boussaïd et al., 2013). One helpful insight for developing heuristics for routing problems is that the problem instances often share common characteristics or patterns, as demonstrated by (Cappart et al., 2021) in a trucking company routing problem where the company needs to generate daily routes for the same city with slight variations due to traffic conditions. These similarities provide opportunities for data-dependent machine learning approaches that can exploit common patterns (Bengio et al., 2021). In recent years, Graph Neural Networks (GNNs) with attention mechanisms have emerged as effective heuristic alternatives for combinatorial optimization problems (Cappart et al., 2021). Leveraging such an approach as a neural heuristic can rapidly generate good paths that utilize the similarities in routing requirements of a Minecraft USAR task. A real-time ASI agent prototype can take advantage of this capability to explore the routing options available to the SAR team under varying conditions.

(Wang et al., 2023) discusses our preliminary explorations on how GNNs with attention mechanisms can be leveraged as neural heuristics to generate good routes for particular states of Minecraft USAR tasks. We build on this earlier work and present a more capable pipeline that utilizes more expressive embeddings and higher-quality training data, resulting in 11% improvement in the length of the generated routes. Furthermore, we perform a new exploratory analysis of the data collected on participant teams performing the Minecraft USAR task and discuss how scrutinizing routing performance can guide the timing of the interventions provided by an ASI agent.

2 BACKGROUND

Combinatorial optimization (CO) is an established interdisciplinary field that has numerous real-world applications, including routing (Korte and Vygen, 2012). Its primary goal is to optimize a cost or objective function by selecting a subset from a finite set while adhering to selection constraints. CO strives to obtain a unique and optimal solution for each problem, but the complexity of certain problems can make this impractical. In such cases, practitioners often rely on problem-specific heuristic methodologies (Boussaïd et al., 2013). However, practical situations frequently involve problem instances that share specific characteristics or patterns (Cappart et al., 2021). These similarities present opportunities for data-dependent machine-learning approaches that can leverage these patterns (Bengio et al., 2021). For example, (Cappart et al., 2021)cite the example of a trucking company's daily routing solutions for the same city, with slight variations in travel times due to varying traffic conditions.

Graph Neural Networks (GNNs) are a potent machine learning architecture that exploits structural, relational, and compositional biases to facilitate geometric deep learning (Gilmer et al., 2017). GNNs aggregate information from structural and feature-based (e.g., node or edge type) graph data into simpler representations of nodes and edges. By parameterizing this aggregation, they can be trained end-to-end against a loss function. GNNs can operate on higher complexity data than what can be represented in regular Euclidean structures, such as an image (2D) or text (1D). GNNs achieve this by being order-invariant, propagating on each node in the graph independently, ignoring the input order, and using the graph structure to guide propagation. These innovations empower GNN models to "reason" about a graph, make general inferences, and use those inferences to make predictions and classifications successfully (Zhou et al., 2020). In recent years, GNNs have been used as neural heuristics to generate solutions for CO problems (Vesselinova et al., 2020). The primary promise of GNNs in this role is that the learned vector representations encode critical graph structures to help solve CO problems more efficiently (Cappart et al., 2021).

In 2018, (Kool et al., 2018) proposed a transformer-like encoder-decoder architecture based on Graph Attention Networks (Veličković et al., 2017) for general routing problems. Their approach trained an encoder-decoder neural network using an actor-critic reinforcement learning approach on randomly generated routing problems. The training did not require optimal solutions to the training instances and could be done in advance. With the trained model, it was possible to generate high-quality solutions to SAR routing problems quickly. Recently, we utilized Kool et al.'s codebase (2018) and augmented it to generate routing solutions for an Artificial Social Intelligence (ASI) agent (Wang et al., 2023). Our task involved a capacitated vehicle routing with-profits model that mapped to the tasks and roles of human participants in a Minecraft USAR mission. Using the trained models, we were able to generate reasonable solutions quickly, informing the ASI agent of potentially good solutions for the given mission state.

2.1 Minecraft USAR Test Environment

One of the main obstacles in creating an agent that can aid human teams is assessing its performance. Prior studies, such as the Electric Elves (Chalupsky et al., 2002), demonstrated the potential of deploying such agents in a real-world research lab. However, they did not quantitatively evaluate their impact on teamwork in the traditional social science sense. Therefore, a significant accomplishment of the ASIST program has been the development of a robust testbed implemented in a game-based (Minecraft) environment, which enables distributed teaming tasks and allows for thorough quantitative evaluations (Freeman et al., 2023). Our approach is to model players, the team, and team processes within the Minecraft environment as Partially Observable Markov Decision Processes (POMDPs). Moreover, our agent incorporates input from analytical components of the testbed as sensors to measure the team and the impact of interventions. Our agent uses behavioral data to inform these POMDP models through Inverse Reinforcement Learning (IRL) and then uses these POMDPs recursively to form a Theory of Mind about the team to reason about the expected utility of alternate interventions. (Pynadath et al., 2023)

The DARPA ASIST Minecraft SAR task environment (Figure 1) is an immersive training platform designed to create Artificial Social Intelligence (ASI) agents. The game's objective is to rescue victims of an urban disaster while earning points. The victims can be either non-critical or critical, and critical victims require a coordinated effort to save. The game environment includes various challenges, such as risks, where a player can get trapped and need assistance from teammates, and rubble, which can impede access to victims. Teams consist of three players who can choose from three roles: medical specialist (medic), hazardous material specialist (engineer), and search specialist (transporter). The medics can triage victims and rescue frozen teammates, the engineers can clear rubble, and the transporters can transport victims. The ASI agents are not embodied team members. Still, they can communicate via text chat with human team members (who can communicate through a shared audio channel).

3 APPROACH

In the Minecraft USAR environment, participants are required to perform a range of tasks across different locations. The primary objective of this environment is to triage victims and move them to secure areas, with the order of victim triage being crucial for mission success due to time constraints. Therefore, the medic role needs to focus on minimizing travel distance, which is similar to the Traveling Salesman's Problem, an NP-Hard optimization problem. However, in this setting, not all relevant locations are known in advance, making it impossible to create an optimal tour before the mission commences. Instead, the ASI agent requires a general framework to quickly generate satisfactory solutions to support or monitor participant performance.

As mentioned, previous work in the ASIST program (Wang et al., 2023) builds on the codebase developed by (Kool et al., 2018) and presents a framework that defines a semantic graph based on Minecraft USAR task maps. This semantic graph captures all the main map entities and structures, including rooms, connections between rooms, victims, and rubble locations, among other features that could be utilized in navigation decisions as depicted in Figure 2. In this representation, each role has different objective nodes; for example, the medic role utilizes the victim nodes, whereas the engineer role requires the rubble nodes that block the victims. To calculate distances between nodes of interest, such as victims, we use Dijkstra's algorithm (Dijkstra, 1959), which considers the layout of the environment. The original framework includes a pipeline to extract information from the maps and convert the resulting distance matrix into a set of 2D coordinates first using Metric Multidimensional Scaling (mMDS) (Kruskal, 1978; Cox and Cox, 2008), also known as Principal Coordinate Analysis (PCoA), and then the John-Lindenstrauss Transform (JLT) (Johnson and Lindenstrauss, 1984) while preserving distance information from the original distance matrix (Indyk et al., 2017). The 2D coordinates are then scaled to $[0,1]$ to fit the original codebase's requirements. The previous work used neural heuristics for route planning on this 2D coordinate set and converted the resulting routes back

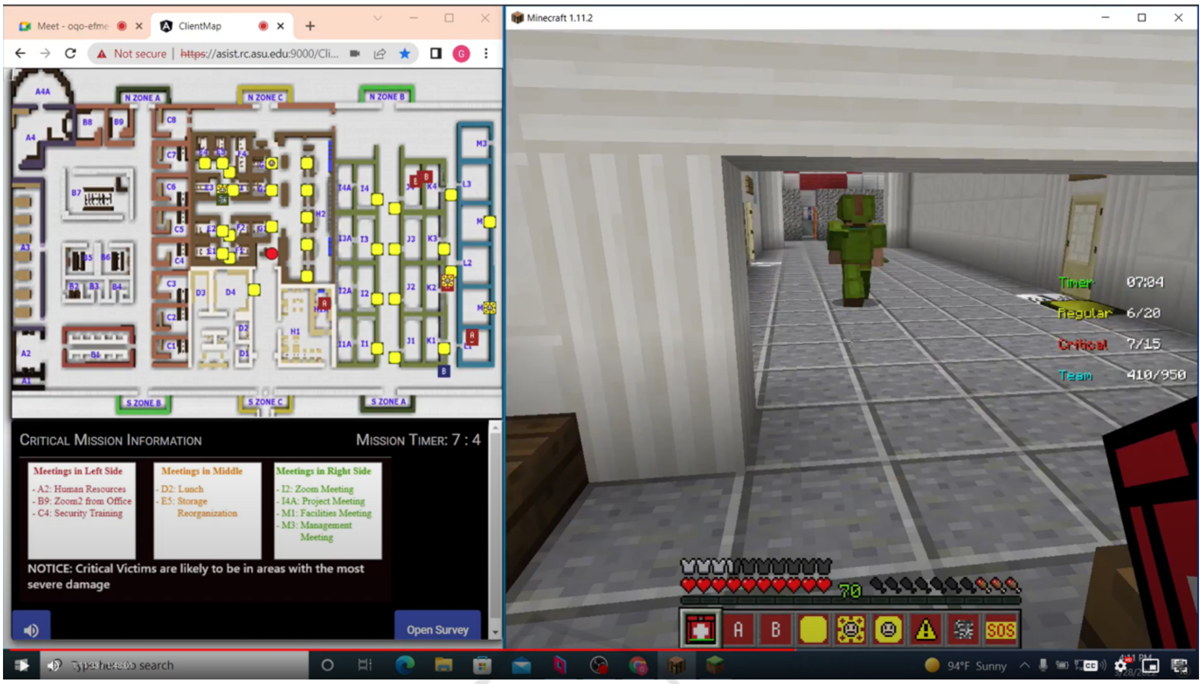

Figure 1: The participant's (medic) interface while playing the Minecraft USAR Scenario.

to the semantic map representation for visualization, analysis, and interfacing purposes.

Our updated framework removes the final step in the pipeline and uses the coordinates in 36D space output by mMDS directly in route planning neural heuristics. This alteration to the pipeline allows for better preservation of the original distance matrix, as converting back to 2D space was causing a loss of precision. The 36D coordinates serve as initial node embeddings for GNN representations. Additionally, we experimented with including victim type as an additional node embedding, given the importance of distinguishing between critical and regular victims for triage.

We simplified the roles of the transporter and engineer by assuming they would follow the medic and perform tasks as required. For example, the transporter transports victims and assists in triaging critical victims while the engineer clears rubble that blocks victims or exits in threat rooms. While a more complex logic could yield potential gains, a simple heuristic in which the engineer and transporter follow the medic was sufficient to generate routes to complete the mission on time. This simplification also streamlines GNN model training by using the total distance traveled by the medic as the primary cost/objective function.

To train the GNN models, we generated 7,000 semantic maps, similar to the number of training instances used in our preliminary explorations, constructed similarly to the Minecraft USAR layouts, with variations in victim locations, threat rooms, and rubble blockages. In addition, we defined ten additional semantic maps as test instances, two of which are the original maps used in the Minecraft USAR task environment.

4 EXPERIMENTS

We trained our model(s) with an actor-critic reinforcement learning approach on the 7,000 randomly generated semantic maps on the base layout. With this approach, we did not supervise the training with the optimal solutions but rather generated gradients based on the best-known solution found so far. We can utilize the trained models in two different ways: (1) As a route generator, where the ASI agent could utilize the model to suggest routes to the team members, and (2) As a team tracker, where the ASI agent could utilize the model to track the team performance with respect to the near-optimal routes generated by the trained model.

4.1 Route Generation

The success of Minecraft USAR missions heavily relies on effective navigation in the Minecraft environment. When the locations of victims are known in advance, minimizing the total mission time for

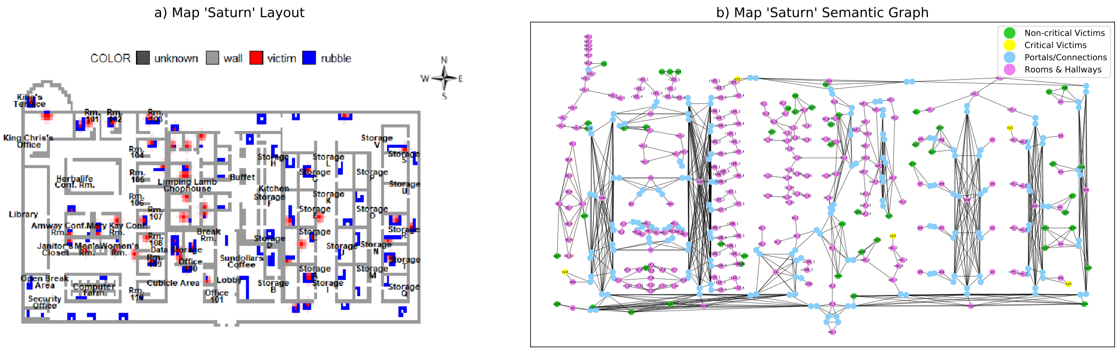

Figure 2: Minecraft USAR Task Map for the Saturn layout and the corresponding Semantic Graph.

the medic is comparable to solving a TSP problem. Therefore, if an ASI agent can access near-optimal routes, it can use them to suggest victim triage plans. We applied this approach to compare the quality of solutions for different configurations, as shown in Table 1. Although all three configurations produce routes that complete the mission within the 1020-second time limit, directly utilizing the 36D node embeddings significantly improves the solution quality, as expected. Moreover, adding victim type as an additional node embedding to our representation leads to slight improvements in solution quality since utilizing victim type information could enhance synchronization among the participants. Finally, the two ground truth layouts used in the Minecraft USAR experiments have solution times of 765 and 770 seconds, respectively, which are well below the 1020-second mark.

Table 1: Mission completion times in seconds for the test set instances.

Model	Average	Median	St. Dev
2D	770.06	767.34	42.16
36D	699	699.12	41.32
36D + Type	685.62	682.66	34.24

Although victim locations may be known in advance, planning the route from scratch may not always be feasible. The team may do things differently than the suggested routes, and hence, the route generation process must consider various starting states. Our solution pipeline can accommodate such changes in initializations. For instance, in a Human Subject Study, we demonstrated that we could run our solution pipeline from any point during the experiment, given the state of the Minecraft USAR task. Furthermore, the generated routing suggestions could enhance the team's performance for the remainder of the mission compared to what had happened in the actual mission. In a particular experiment, for example, a human subject team's performance in scored game points could have improved by 30 % if they followed the routing suggestion generated by our pipeline for the rest of the mission at the 2/3rd mark.

However, in both the Minecraft USAR environment and real-life USAR missions, it is typically not feasible to know the locations of victims in advance. Therefore, the actual use case is more complex, even with real-time near-optimal route planning capability. For instance, in the Minecraft USAR environment, each participant is provided unique knowledge based on their roles, including information about potential victim locations, threat rooms, sensory cues to assist in finding victims, and more. In Human Subject Experiments, participants begin the scenario with this knowledge and simultaneously execute the assigned search and rescue task while exploring the environment. When we initiated our solution pipeline with the same knowledge and only planned for the known artifacts (but perceived additional information during the execution of the planned routes), the model located and triaged 85% of the victims in around 660 seconds, leaving enough time for further exploration. However, we did not implement an exploration algorithm to augment our model for this task version.

4.2 Team Tracking

An ASI agent participating in the DARPA ASIST program is expected to facilitate better teamwork through interventions. Although there are various approaches to making such interventions, monitoring the navigation performance of a team and identifying significant shifts in this performance can be a valuable indicator for the ASI agent. In the Minecraft USAR environment described by (Freeman et al., 2023), a Human Subject Research study was carried out to evaluate the performance of ASI agents. The study involved 242 trials with different configurations, and our analysis focuses on 238 of these trials (due to data issues with the remaining 4). Our investigation aims to determine whether the neural heuristics' near-optimal routes can be utilized to track the navigation performance of human teams.

Each trial includes a metadata file that captures the Minecraft environment state during the mission. At mission times 0, 3, 5, 8, and 11 minutes, we took snapshots of the environment state and used our solution pipeline to generate solutions for the remaining tasks based on that specific state. It is important to note that our solution pipeline only focuses on unfinished tasks and does not attempt to complete tasks that have already been completed, such as triaging a victim. This approach allowed us to gather a list of lower bounds on mission completion times based on the progress made by the team up to each specific mission time. We then used these hypothetical completion times to calculate the deviation from the near-optimal route that was calculated at time 0, which represents a complete solution generated with perfect information. By tracking these deviations over time, we can detect significant changes in the team's navigation performance and identify opportunities for intervention.

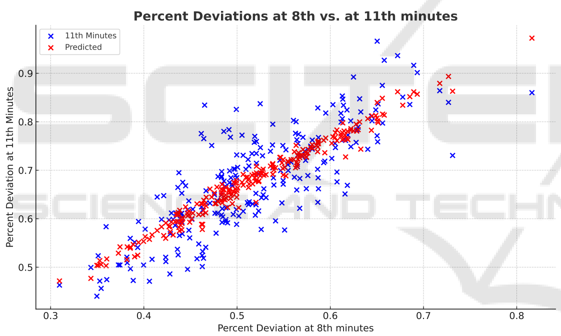

Figure 3: Multivariable linear regression analysis of percent deviations from the optimal route.

To analyze the data, we used multivariable linear regression, and even with the inherent configuration variations in each trial, there is a trendline as shown in Figure 3. We expect that for most teams, the deviation from the optimal route will increase over time, following the trendline. However, major deviations from this trendline would indicate a significant shift in the team's navigation performance, which would warrant intervention.

To identify major deviations from the expected trendline, we can examine the standard residual plots of the multivariable linear regression model, as shown in Figure 4. In particular, any standard residual above 2 (represented by the red dots on the plot) would warrant further investigation to determine whether an intervention is needed to improve the team's performance. Conversely, standard residuals less than -2 (represented by the green dots on the plot) could indicate areas where the team's navigation performance can be improved, providing valuable insights into their overall performance.

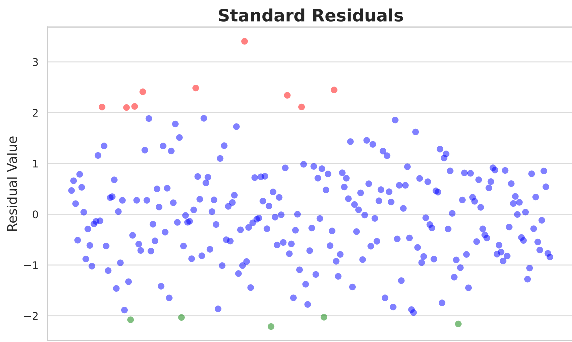

Figure 4: Standard residuals from the multivariable linear regression model of deviations from optimal route.

5 DISCUSSION

We have significantly improved the existing solution pipeline, resulting in better solution quality and demonstrating its potential as a reliable tool for a USAR ASI agent. However, it's important to note that the pipeline's performance superior to human participant teams was expected to some extent due to its access to maps and victim locations. Nonetheless, Graph Neural Networks offer a robust architecture for developing neural heuristics for combinatorial problems like routing, particularly when there is significant similarity among problem instances. Graph Neural Networks' flexibility and ability to adapt to changes in objective functions make them an appropriate choice for handling additional problem information through node embeddings. This flexibility, along with the ability to rapidly generate routing solutions under different conditions, including starting with limited information similar to human participant teams, allows the ASI agent to better evaluate the routes available to a USAR team.

To speed up training while still generating quality solutions, we made the assumption in our solution pipeline that the engineer and the transporter would follow the medic, as we found that using a more insightful heuristic for routing them did not result in any improvements in our exploratory experiments. However, we acknowledge that this assumption may not be the most effective for an actual deployment, and further improvements may be necessary to optimize the use of the transporter and the engineer.

Although we believed that monitoring the navigation performance of participants and detecting deviations would provide valuable insights for the ASI agent on the state of the teams, we were unable to test these ideas due to budget cuts. Furthermore, in our exploratory analyses, we did not have enough information to determine whether teams with high standard residual values had actual problems. Thus, our results are mainly exploratory, and further investigation is needed to assess the effectiveness of our proposed approach.

6 CONCLUSIONS

We have developed a versatile routing system that utilizes neural heuristics to efficiently guide a real-time ASI agent on available routing options for a USAR team based on the current state of the mission. This system can serve as a reliable tool for the ASI agent to analyze routing options for the USAR team it is assisting. Additionally, the framework enables the ASI agent to monitor the team's navigation performance and identify any potential difficulties they may be experiencing. By leveraging this information along with other insights, detecting such issues can prompt effective interventions by the ASI agent.

ACKNOWLEDGEMENTS

Part of the effort depicted is sponsored by the U.S. Army Research Laboratory (ARL) under contract number W911NF-14-D-0005 and by the Defense Advanced Research Projects Agency (DARPA) under contract number W911NF2010011, and that the content of the information does not necessarily reflect the position or the policy of the Government or the Defense Advanced Research Projects Agency, and no official endorsements should be inferred.

REFERENCES

Bengio, Y., Lodi, A., and Prouvost, A. (2021). Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421.

Boussaïd, I., Lepagnot, J., and Siarry, P. (2013). A survey on optimization metaheuristics. *Information sciences*, 237:82–117.

Cappart, Q., Chételat, D., Khalil, E., Lodi, A., Morris, C., and Veličković, P. (2021). Combinatorial optimization and reasoning with graph neural networks. *arXiv preprint arXiv:2102.09544*.

Chalupsky, H., Gil, Y., Knoblock, C. A., Lerman, K., Oh, J., Pynadath, D. V., Russ, T. A., and Tambe, M. (2002). Electric elves: Agent technology for supporting human organizations. *AI magazine*, 23(2):11–11.

Cox, M. A. and Cox, T. F. (2008). Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.

Freeman, J., Huang, L., Wood, M., and Cauffman, S. J. (2023). Evaluating artificial social intelligence in an urban search and rescue task environment. In *Computational Theory of Mind for Human-Machine Teams: First International Symposium, ToM for Teams 2021, Virtual Event, November 4–6, 2021, Revised Selected Papers*, pages 72–84. Springer.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR.

Indyk, P., Matoušek, J., and Sidiropoulos, A. (2017). 8: low-distortion embeddings of finite metric spaces. In *Handbook of discrete and computational geometry*, pages 211–231. Chapman and Hall/CRC.

Johnson, W. B. and Lindenstrauss, J. (1984). Extensions of lipschitz mappings into a hilbert space 26. *Contemporary mathematics*, 26.

Kool, W., Van Hoof, H., and Welling, M. (2018). Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*.

Korte, B. and Vygen, J. (2012). The traveling salesman problem. In *Combinatorial Optimization*, pages 557–592. Springer.

Kruskal, J. B. (1978). *Multidimensional scaling*. Number 11. Sage.

Pynadath, D., Gurney, N., Kenny, S., Kumar, R., Marsella, S., Matuszak, H., Mostafa, H., Sequeira, P., Ustun, V., and Wu, P. (2023). Effectiveness of teamwork-level interventions through decision-theoretic reasoning in a minecraft search-and-rescue task. In *Proceedings of the 2023 international conference on autonomous agents & multiagent systems*.

Sukthankar, G., Sycara, K., Giampapa, J. A., Burnett, C., and Preece, A. (2007). Towards a model of agent-assisted team search. In *Procceedings of the First Annual Conference of the International Technology Alliance in Network and Information Science*.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Vesselinova, N., Steinert, R., Perez-Ramirez, D. F., and Boman, M. (2020). Learning combinatorial optimization on graphs: A survey with applications to networking. *IEEE Access*, 8:120388–120416.

Wang, Y., Gurney, N., Zhou, J., Pynadath, D. V., and Ustun, V. (2023). Route optimization in service of a search and rescue artificial social intelligence agent. In *Computational Theory of Mind for Human-Machine Teams: First International Symposium, ToM* *for Teams 2021, Virtual Event, November 4–6, 2021, Revised Selected Papers*, pages 220–228. Springer.

Webber, S. S., Detjen, J., MacLean, T. L., and Thomas, D. (2019). Team challenges: Is artificial intelligence the solution? *Business Horizons*, 62(6):741–750.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81.