

pyMassEvac: A Python package for simulating multi-domain mass evacuation operations

Mark Rempel ^{1¶}

¹ Defence Research and Development Canada, Canada ¶ Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

pyMassEvac is a Python package whose aim is to study mass evacuation scenarios. In particular, it is designed to simulate single- and multi-domain mass evacuation operations in which:

- the individuals to be evacuated are at a remote location, such as in the Arctic, where access to immediate medical care is limited or non-existent;
- each individual's medical condition may change over time, perhaps due to environmental conditions, injury, or care being provided; and
- the individuals must be transported from the evacuation site to a Forward Operating Location (FOL).

An example of a multi-domain mass evacuation operation, where the objective is to maximize the number of lives saved by transporting individuals to an FOL, is depicted in [Figure 1](#).

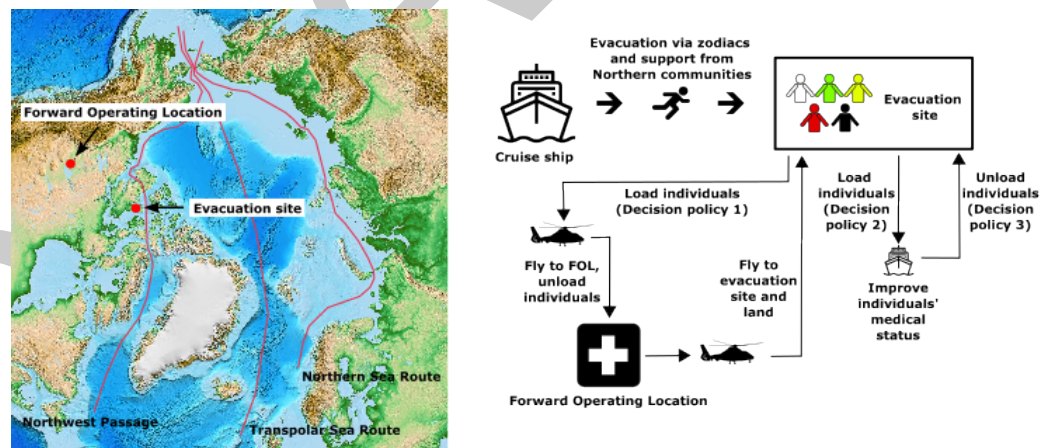


Figure 1: Evacuation plan via air with medical assistance provided at the evacuation site via ship. Colours of individuals at the evacuation site represent those in different triage categories (white, green, yellow, red, black; black represents deceased). For a full description, see Rempel (2024).

Within this context, pyMassEvac may be used to provide decision support to defence and security planners in two ways. First, through exploring the impact of policies used to make the three decisions depicted in [Figure 1](#) (see right panel):

- **Decision policy 1:** the policy that determines which individuals are loaded onto a vehicle, such as a helicopter, for transport from the evacuation site to the FOL;
- **Decision policy 2:** the policy that determines which individuals receive medical care (if available) at the evacuation site, such as onboard a nearby ship; and

23 ▪ **Decision policy 3:** the policy that determines which individuals are removed from the
24 group receiving medical care, for reasons such as limited capacity or that the individuals'
25 medical condition has been sufficiently improved, and returned to the group ready to be
26 transported to the FOL.

27 Second, assuming decision policies are selected, decision support may be provided by using
28 pyMassEvac to explore the selected policies' robustness to changes in a scenario's parameters.
29 For example, pyMassEvac may be used to explore how robust a set of decision policies are in
30 terms of the number of lives saved with respect to:

- 31 ▪ the initial arrival time of one or more transport vehicles after the individuals have arrived
32 at the evacuation site;
- 33 ▪ the travel time between the evacuation site and the FOL; and
- 34 ▪ the rate at which an individual's medical condition becomes better (through receiving
35 medical care) or worse (due to injury or exposure to environmental conditions) over time.

36 Changes in a scenario's parameters from baseline values may reflect a variety of real-world
37 strategic and operational decisions beyond the tactical decisions made within scenario itself.
38 For example:

- 39 ▪ the reduction in the initial arrival time of transport vehicles may reflect an operational
40 decision to pre-position vehicles during the summer season;
- 41 ▪ the reduction in the travel time between the evacuation site and FOL may reflect a
42 strategic decision to build a new aerodrome; and
- 43 ▪ the decrease in the rate at which an individual's medical condition worsens may reflect
44 an operational decision to invest in improved medical kit.

45 Thus, pyMassEvac is designed to be primarily used by operational researchers who study
46 humanitarian or defence and security operations.

47 pyMassEvac is accessible at <https://github.com/mrempel/pyMassEvac> and is installed via
48 a setup.py script. In addition, published evacuation scenarios that have studied using this
49 package (or one of its earlier developmental versions) are described in Rempel et al. (2021),
50 Rempel & Shiell (2023), and Rempel (2024).

51 Statement of need

52 The significant decrease in Arctic sea ice in recent decades has resulted in increased activity in
53 the Arctic across a range of sectors, such as oil and gas, mining, fishing, and tourism. With
54 respect to tourism and the potential increase in the number of Arctic cruise ships, Arctic
55 nations are concerned with both the potential increase in the number of Search and Rescue
56 (SAR) incidents that may occur and the increased size of those incidents in terms of the
57 number of individuals in need of evacuation. This is evidenced by recent exercises that have
58 been conducted, such as the SARex series in Norway (Solberg et al., 2016, 2018), a table-top
59 exercise including the United States, Canada, and the cruise ship industry (McNutt, 2016),
60 and NANOOK-TATIGIT 21 by the Canadian Armed Forces (National Defence, 2021).

61 While software exists to support planning for and executing evacuation operations, this software
62 either requires a paid license (AVN, 2025; SAR Technology Inc., 2025), focuses on search
63 planning (United States Coast Guard, 2025), or addressed specific situations such as wildfires
64 (Guman et al., 2024). With this in mind, pyMassEvac aims to provide an open source software
65 package that enables researchers to both assess the impact of strategic and operational decisions
66 made prior to an evacuation operation occurring, as well as the impact of tactical decisions
67 made within the operation itself.

Features

Defining an evacuation operation

Mass evacuation operations are modelled in pyMassEvac as a sequential decision problem under uncertainty using Powell's universal framework for sequential decisions (Powell, 2022). See Section 4 of Rempel (2024) for the complete description of the model. Given this framework, a scenario's parameters are specified via the initial state variable S_0 , which consists of the following elements:

- m^e : Vector of mean time (hours) for an individual's medical condition to worsen and transition from a triage category $t \in \mathcal{T} \setminus \{b\}$ to the next lower triage category $t' \in \mathcal{T} \setminus \{w\}$ at the evacuation site, i.e., m_w^e is the mean transition time from the white (w) to green (g) tag category. The set of triage categories is given as $\mathcal{T} = \{w, g, y, r, b\}$;
- m^s : Vector of mean time (hours) for an individual's medical condition to improve and transition from a triage category $t \in \mathcal{T} \setminus \{w, b\}$ to the next higher triage category $t' \in \mathcal{T} \setminus \{r, b\}$ while receiving medical care, i.e., m_r^s is the mean transition time from the red (r) to yellow (y) tag category;
- c^h : Total capacity for individuals onboard a transport vehicle, such as a helicopter;
- c^s : Total capacity for individuals to receive medical care, such as onboard a ship;
- δ^h : Vector of capacity consumed by each triage category $t \in \mathcal{T} \setminus \{b\}$ onboard a transport vehicle. Individual in the black (b) tag category are not transported as they are deceased and are assumed to be recovered at the end of the rescue operation;
- δ^s : Vector of capacity consumed by each triage category $t \in \mathcal{T} \setminus \{b\}$ when receiving medical care;
- η^h : Total time (hours) for a transport vehicle to load individuals at the evacuation site, transport them to the FOL, unload the individuals, and return to the evacuation site;
- η^{sl} : Total time (hours) to transfer individuals at the evacuation site to the local facility (such as a ship) in which they will receive medical care, plus the time until a decision is made as to which individuals to transfer back to the evacuation site;
- η^{su} : Total time to transfer individuals from the local facility (such as a ship) in which they are receiving medical care to the evacuation site, plus the time until a decision is made as to which individuals to transport to the FOL;
- τ^h : Vector of initial arrival time (hours) of each transport vehicle after the individuals have arrived at the evacuation site; and
- τ^s : Vector of initial arrival time (hours) of each medical care facility (such as a ship) after the individuals have arrived at the evacuation site.

Note that the initial state in pyMassEvac differs from Rempel (2024) as this package includes both τ^h and τ^s . In Rempel (2024) these two parameters were specified separately in the case study presented in Section 5.

An example of an initial state, with one transport vehicle and one medical care facility, is given in the tutorial found in `tutorial\tutorial.ipynb`.

Example decision policies

pyMassEvac provides a set of decision policies that implements those described in Rempel (2024). All policies are defined in `mass_evacuation_policy.py` and are summarized as follows:

- `green_first_loading_policy`: This policy may be used for either **Decision policy 1** or **Decision policy 2** and puts an emphasis on loading healthier individuals prior to those with worse medical conditions;
- `yellow_first_loading_policy`: This policy is similar to the green-first loading policy, with the exception that it focuses on those individuals that require near-term care, followed by those in descending order in triage category. This policy may be used for either **Decision policy 1** or **Decision policy 2**;

- `critical_first_loading_policy`: This policy prioritizes those individuals that require immediate attention before moving onto less critical categories. This policy may be used for either **Decision policy 1** or **Decision policy 2**;
- `random_loading_policy`: This policy randomly selects individuals, regardless of their triage category. This policy may be used for either **Decision policy 1** or **Decision policy 2**;
- `random_unloading_policy`: This policy randomly selects individuals, regardless of their triage category. This policy may be used for **Decision policy 3**; and
- `white_unloading_policy`: This policy only removes individuals from the medical facility whose medical condition has improved such that they are assigned a white (w) tag. This policy may be used for **Decision policy 3**.

In addition, a `do_nothing` policy is provided to model situations in which a decision is to be delayed or a model the lack of transport of medical care.

The tutorial found in `tutorial\tutorial.ipynb` demonstrates how to use these decision policies. Specifically, it uses the `green_first_loading_policy` for **Decision policy 1** and **Decision policy 2**, and the `white_unloading_policy` for **Decision policy 3**.

Ready for reinforcement learning

`pyMassEvac` is implemented as a custom Gymnasium environment (Towers et al., 2024). An example of its use as an environment with fixed decision policies is provided in `tutorial\tutorial.ipynb`. `pyMassEvac` may also be used in combination with a reinforcement learning to seek optimal, or at least near-optimal, decision policies. Among the many considerations that must be made when selecting or designing a learning algorithm for this environment is that the set of valid actions are dependent on both the state variable S_k and the parameters defined in the initial state S_0 —see Section 4.1 of Rempel (2024). Thus, when using a reinforcement learning algorithm a form of invalid action masking (Hou et al., 2023; Huang & Ontañón, 2022) must be implemented.

Acknowledgements

I acknowledge contributions from both Nicholi Shiell and Kaeden Tessier, who are co-authors on related papers. These collaborations inspired the development of this package.

References

- AVN. (2025). *MassEvac*. <https://avncorp.com/projects/massevac/>
- Guman, J., O'Brien, J., Pondoc, C., & Kochenderfer, M. (2024). PyroRL: A reinforcement learning environment for wildfire evacuation. *Journal of Open Source Software*, 9(101), 6739. <https://doi.org/10.21105/joss.06739>
- Hou, Y., Liang, X., Zhang, J., Yang, Q., Yang, A., & Wang, N. (2023). Exploring the use of invalid action masking in reinforcement learning: A comparative study of on-policy and off-policy algorithms in real-time strategy games. *Applied Sciences*, 13(14). <https://doi.org/10.3390/app13148283>
- Huang, S., & Ontañón, S. (2022). A closer look at invalid action masking in policy gradient algorithms. *The International FLAIRS Conference Proceedings*, 35. <https://doi.org/10.32473/flairs.v35i.130584>
- McNutt, C. (2016). *Northwest Passage (NWP 16) 2016 Exercise—after action report*. <https://www.hSDL.org/c/abstract/?docid=802138>

- 160 National Defence. (2021). *Department of National Defence and Canadian Armed Forces*
161 *2021-22 Departmental Plan*. [https://www.canada.ca/content/dam/dnd-mdn/documents/](https://www.canada.ca/content/dam/dnd-mdn/documents/departmental-results-report/2021-2022/departmental-plan-2021-22.pdf)
162 [departmental-results-report/2021-2022/departmental-plan-2021-22.pdf](https://www.canada.ca/content/dam/dnd-mdn/documents/departmental-results-report/2021-2022/departmental-plan-2021-22.pdf)
- 163 Powell, W. (2022). *Reinforcement learning and stochastic optimization: A unified framework*
164 *for sequential decisions* (1st ed.). Wiley.
- 165 Rempel, M. (2024). Modelling a major maritime disaster scenario using the universal modelling
166 framework for sequential decisions. *Safety Science*, 171, 106379. [https://doi.org/10.1016/](https://doi.org/10.1016/j.ssci.2023.106379)
167 [j.ssci.2023.106379](https://doi.org/10.1016/j.ssci.2023.106379)
- 168 Rempel, M., & Shiell, N. (2023). Using reinforcement learning to provide decision support
169 in multi-domain mass evacuation operations. *NATO STO Review, Fall 2023*, 21-1-21-20.
170 <https://review.sto.nato.int/>
- 171 Rempel, M., Shiell, N., & Tessier, K. (2021). An approximate dynamic programming approach
172 to tackling mass evacuation operations. *2021 IEEE Symposium Series on Computational*
173 *Intelligence (SSCI)*, 01–08. <https://doi.org/10.1109/SSCI50451.2021.9659974>
- 174 SAR Technology Inc. (2025). *Incident Commander Pro*. [https://sarotechnology.ca/](https://sarotechnology.ca/sarotechnology/sarotechnology.htm/#ICProV7)
175 [sarotechnology/sarotechnology.htm/#ICProV7](https://sarotechnology.ca/sarotechnology/sarotechnology.htm/#ICProV7)
- 176 Solberg, K. E., Gudmestad, O. T., & Bjarte, B. O. (2016). *SARex Spitzbergen: Search and*
177 *rescue exercise conducted off North Spitzbergen: Exercise report* (Report 58). University
178 Stravanger.
- 179 Solberg, K. E., Gudmestad, O. T., & Bjarte, B. O. (2018). *SARex3 Evacuation to shore,*
180 *survival and rescue* (Report 75). University Stravanger.
- 181 Towers, M., Kwiatkowski, A., Terry, J., Balis, J. U., Cola, G. D., Deleu, T., Goulão, M.,
182 Kallinteris, A., Krimmel, M., KG, A., Perez-Vicente, R., Pierré, A., Schulhoff, S., Tai, J.
183 J., Tan, H., & Younis, O. G. (2024). *Gymnasium: A standard interface for reinforcement*
184 *learning environments*. <https://arxiv.org/abs/2407.17032>
- 185 United States Coast Guard. (2025). *Search and rescue optimal planning system (SAROPS)*.
186 [https://www.dcms.uscg.mil/Our-Organization/Assistant-Commandant-for-Acquisitions-CG-9/](https://www.dcms.uscg.mil/Our-Organization/Assistant-Commandant-for-Acquisitions-CG-9/International-Acquisition/SAROPS/)
187 [International-Acquisition/SAROPS/](https://www.dcms.uscg.mil/Our-Organization/Assistant-Commandant-for-Acquisitions-CG-9/International-Acquisition/SAROPS/)