

Adaptação de Soluções Logísticas Multi-Escalão em Simulações Baseadas em Agentes

Davi C. Machado, Murilo H. Block, Luis A. L. Silva

¹Curso de Ciência da Computação, Universidade Federal de Santa Maria
Av. Roraima nº 1000, Santa Maria – RS, CEP: 97105-900

{ddmachado, mhblock, luisalvaro}@inf.ufsm.br

Abstract. *This work proposes a hierarchical adaptation method for logistic solutions employed in the automation of agent behaviors within simulation-based training environments. The approach combines Case-Based Reasoning (CBR) with multilevel adaptation rules that adjust plans retrieved from case bases to the specific conditions of simulated logistic problems. The modeled adaptation rules take into account doctrinal criteria, operational and tactical situations, and contextual constraints analyzed from the current simulation status, ensuring coherence among procedures executed by different logistic echelons.*

Resumo. *Este trabalho propõe um método de adaptação hierárquica de soluções logísticas utilizadas na automatização de comportamentos de agentes em ambientes de treinamento baseado em simulação. A proposta une Raciocínio Baseado em Casos (CBR) e regras de adaptação multinível, as quais ajustam planos logísticos recuperados de bases de casos a condições específicas de problemas simulados. As regras de adaptações modeladas consideram critérios doutrinários, situações operacionais e táticas e restrições contextuais analisadas a partir do status atual das simulações, garantindo coerência entre procedimentos executados por diferentes escalões logísticos.*

1. Introdução

Sistemas de simulação virtuais táticos oferecem ambientes desafiadores para o desenvolvimento de algoritmos de inteligência artificial (IA). No nosso projeto, agentes simulados devem atuar de forma adaptativa para executar missões que requerem a execução de diferentes procedimentos logísticos. Entre outras técnicas de IA aplicáveis para abordar esses problemas, raciocínio baseado em casos (Case-Based Reasoning — CBR) [Aamodt and Plaza 1994] explora a resolução de problemas por meio da reutilização de soluções passadas. Essa abordagem de IA é especialmente relevante em domínios de problemas onde experiências de soluções de problemas são essenciais para o desenvolvimento de atividades de treinamento baseadas em simulação [Pozzer et al. 2022].

[Oh et al. 2017] investiga o uso de técnicas de imitação e CBR para reproduzir comportamentos (*micromanagement*) em jogos de RTS, adaptando ações de agentes a partir de comportamentos humanos registrados. [Liu et al. 2016] propõe uma abordagem com algoritmos genéticos para evoluir micromovimentos, otimizando parâmetros táticos das unidades. [Rogers and Skabar 2014] apresenta o *Micromanagement Task Allocation System* (MTAS), que emprega alocação multinível de tarefas e tomada de decisão.

[de Freitas et al. 2024] introduz uma arquitetura para automatizar, de forma realista e conforme doutrina militar, comportamentos de agentes, interpretando scripts externos que são convertidos em comandos de simulação. Neste contexto de pesquisa, o foco do processo de micromanagement investigado neste trabalho é o reuso e adaptação hierárquica de experiências de solução de problemas logísticos simulados.

Problemas logísticos de (re)suprimento em simulações táticas executadas no nosso projeto são abordados de acordo com diferentes níveis hierárquicos, ou escalões logísticos. Desta forma, o trabalho apresenta um método hierárquico de adaptação de comportamentos logísticos simulados. O trabalho explora a modelagem e implementação de um conjunto de regras de adaptação. Essas regras são aplicadas a comandos de simulação modelados em scripts logísticos. Com base em métricas como custo operacional e similaridade com planos táticos humanos construídos para problemas usados como consulta no sistema, experimentos desenvolvidos demonstram que o modelo proposto permite construir soluções eficientes e consistentes com procedimentos logísticos desenvolvidos no mundo real.

2. Um método hierárquico de adaptação de soluções logísticas

O método proposto é baseado na representação, recuperação, adaptação e reuso hierárquico de soluções de problemas. Casos de solução de problemas foram modelados a partir de soluções elaboradas e executadas por usuários de um simulador. Estes casos capturam experiências concretas de solução de problemas logísticos simulados. Cada caso é representado por atributos que contêm características da missão logística atual a ser executada, descrevendo tipo e quantidade de suprimento, status e localização de agentes, e outros aspectos operacionais e táticos do problema. A solução de problemas é registrada no corpo dos casos como um script textual (Tabela 1). Esse script lista a sequência de comandos de simulação necessários para resolver o problema.

O processo de recuperação de casos utiliza o algoritmo K-NN. A similaridade entre casos é medida por funções de similaridade modeladas para os diferentes tipos de atributos usados na representação de casos. Durante as simulações, o caso mais similar a um problema logístico atual é recuperado de uma base de casos. Três diferentes bases de casos contendo soluções para problemas logísticos em diferentes escalões foram modeladas neste trabalho. As adaptações executadas em soluções recuperadas variam desde ajustes pontuais em procedimentos de (re)suprimento de agentes individuais até reestruturações mais amplas de sequências de ações executadas por grupos de agentes simulados. O Algoritmo 1 detalha o método de adaptação implementado.

Na análise das soluções recuperadas e adaptadas, exemplificadas na Tabela 1, as seguintes adaptações foram realizadas: remoção, substituição e manutenção. No nível Escalão Superior, a adaptação preservou a posição tática Bravo, mas alterou o conjunto de agentes fonte e removeu a participação do agente 11 da missão logística. Considerando comandos de (re)ssuprimento e deslocamento usados nas simulações, o agente $Ag=7$ foi substituído por $Ag=6$, e o suprimento $Sup=1$ foi substituído por $Sup=2$. Apesar dessas alterações, as quantidades de suprimento transferidas e os destinos táticos definidos (Bravo e Alpha) foram mantidos, bem como a ordem de execução dos comandos de simulação. Em síntese, o processo de adaptação reconfigurou agentes e suprimentos, mantendo as quantidades de suprimentos e posições táticas, ao mesmo tempo que elimi-

nou uma ação logística de nível superior não necessária. O resultado é um script ajustado para cumprir a missão logística determinada nas simulações. Este plano foi adaptado à disponibilidade de agentes e recursos analisados no estado atual das simulações.

Algorithm 1 Método de Adaptação Hierárquica de Soluções Logísticas

Require: ($aLogQuery, LogEchelons, AllCaseBases, simThreshold, AdaptRules$) ▷ Parâmetros e componentes do problema logístico
Ensure: $adaptedLogScript$ ▷ Script logístico adaptado

```

1: for all  $aEchelon \in LogEchelons$  do ▷ Percorre os escalões logísticos modelados
2:    $aLogCaseBase \leftarrow getCaseBase(aEchelon, AllCaseBases)$  ▷ Obtém a base de casos do escalão
3:    $aEchelonQuery \leftarrow getEchelonQuery(aLogQuery)$  ▷ Focaliza consulta no escalão
4:    $aListOfLogCases \leftarrow \{ \{logCase_1, \dots, logCase_n\} \in aLogCaseBase \mid$ 
      $similarity(aEchelonQuery, logCase_i) \geq simThreshold \}$  ▷ Recupera casos com similaridade desejada
5:    $aLogCase \leftarrow getLogCase(first, aListOfLogCases)$  ▷ Seleciona o caso mais similar
6:    $aLogScript \leftarrow getLogScript(aLogCase)$  ▷ Extrai o script logístico do caso
7:   for all  $scriptAction \in aLogScript$  do ▷ Percorre todas as ações do script
8:     if  $isActionRequired(aLogScript, aLogQuery)$  then ▷ Verifica se a ação é necessária
9:        $aLogScript \leftarrow adapt(getTransp(scriptAction, aLogQuery, AdaptRules))$  ▷ Adapta transporte suprimento
10:       $aLogScript \leftarrow adapt(getSupply(scriptAction, aLogQuery, AdaptRules))$  ▷ Adapta o tipo de suprimento
11:       $aLogScript \leftarrow adapt(getSupplyAmount(scriptAction, aLogQuery, AdaptRules))$  ▷ Adapta a qtd de suprimento
12:       $aLogScript \leftarrow adapt(getSourceAgent(scriptAction, aLogQuery, AdaptRules))$  ▷ Adapta o agente de origem
13:       $aLogScript \leftarrow adapt(getTargetAgent(scriptAction, aLogQuery, AdaptRules))$  ▷ Adapta o agente de destino
14:       $aLogScript \leftarrow adapt(getTacticalPosition(scriptAction, aLogQuery, AdaptRules))$  ▷ Adapta a posição tática
15:     else
16:        $aLogScript \leftarrow remove(scriptAction, aLogScript, AdaptRules)$  ▷ Remove ações desnecessárias
17:     end if
18:   end for
19:   while  $isSupplyRequired(aLogScript, aLogQuery)$  do ▷ Enquanto houver necessidade de suprimento
20:      $scriptAction \leftarrow build(aLogScript, aLogQuery)$  ▷ Cria nova ação logística
21:      $aLogScript \leftarrow add(scriptAction, aLogScript)$  ▷ Adiciona ação ao script
22:   end while
23:    $adaptedLogScript \leftarrow append(aLogScript, adaptedLogScript)$  ▷ Anexa script adaptado ao resultado final
24: end for
25: return  $adaptedLogScript$  ▷ Retorna o script adaptado completo

```

Tabela 1. Scripts base representados em casos recuperados para consultas executadas em múltiplos escalões logísticos e scripts adaptados para atender aos requisitos e restrições da missão logística corrente.

Nível	Soluções recuperadas	Soluções adaptadas
Escalão Superior	Deslocar(Ag=12, {(Sup=1, Q=12), (Sup=2, Q=12)}, Pos=Bravo) Deslocar(Ag=11, {(Sup=3, Q=24)}, Pos=Bravo)	Deslocar(Ag=12, {(Sup=2, Q=12), (Sup=3, Q=12)}, Pos=Bravo)
Grupo	PlanRessup(Ag=12 → Ag=7, {(Sup=1, Q=8)}, Pos=Bravo) ExecutarRessup() Deslocar(Ag=7, {(Sup=1, Q=8)}, Pos=Alpha)	PlanRessup(Ag=12 → Ag=6, {(Sup=2, Q=8)}, Pos=Bravo) ExecutarRessup() Deslocar(Ag=6, {(Sup=2, Q=8)}, Pos=Alpha)
Unidade	PlanRessup(Ag=7 → Ag=2, {(Sup=1, Q=4)}, Pos=Alpha) PlanRessup(Ag=7 → Ag=1, {(Sup=1, Q=4)}, Pos=Alpha) ExecutarRessup()	PlanRessup(Ag=6 → Ag=2, {(Sup=2, Q=4)}, Pos=Alpha) PlanRessup(Ag=6 → Ag=1, {(Sup=2, Q=4)}, Pos=Alpha) ExecutarRessup()

3. Experimentos e Resultados

O desempenho do método hierárquico de reuso e adaptação de soluções proposto foi avaliado por meio de testes onde soluções geradas e soluções criadas por usuários das simulações foram comparadas. Os testes foram conduzidos em 37 novos problemas logísticos não vistos durante o desenvolvimento do sistema e utilizaram bases de casos com 450 problemas e soluções logísticas de 3 diferentes escalões simulados.

Uma métrica de custo logístico foi adotada para a comparação de soluções, definida pela soma dos custos individuais das ações de simulação que compõem um plano logístico. Esses custos consideram o tipo de agente, a operação realizada, as distâncias percorridas e os suprimentos movimentados. Nos experimentos, estabeleceu-se uma margem de tolerância segundo a qual soluções com diferença de até 3% no custo logístico

foram tratadas como equivalentes. Além disso, o processo de recuperação de casos utilizou um limiar de similaridade de 90%. A avaliação de desempenho do sistema foi conduzida por meio de um critério de maioria: para cada problema, até 10 soluções adaptadas eram geradas, sendo cada uma considerada um “voto positivo” caso o custo logístico permanecesse dentro da margem de 3% em relação à solução humana; o sistema era então considerado bem-sucedido se mais da metade das soluções produzidas se qualificasse como votos positivos.

Os resultados dos testes são os seguintes: Acurácia = 0,946, Precisão = 1,000, Recall = 0,946, F1-Score = 0,972. Esta análise revelou a capacidade do sistema de otimizar custos: 117 soluções (36,3%) apresentaram um custo menor que a solução humana, 159 soluções (49,4%) apresentaram um custo igual ao da solução humana, 46 soluções (14,3%) apresentaram um custo maior (dentro da margem de tolerância). Estes números demonstram que, além de replicar soluções construídas por usuários das simulações, o método proposto é capaz de propor alternativas logísticas eficientes.

4. Conclusões

Este trabalho investiga o reuso de soluções de (re)suprimento em ambientes de treinamento baseados em simulações. O trabalho propõe uma abordagem baseada em CBR com adaptação hierárquica multinível, capaz de ajustar planos recuperados às condições específicas de cada problema. O modelo hierárquico de adaptação de casos demonstrou ser eficaz em automatizar o comportamento de agentes envolvidos nas simulações, oferecendo soluções que se aproximam de procedimentos logísticos multi-escalon construídos por usuários das simulações. Resultados de testes evidenciam os ganhos da proposta em termos de custos logísticos em comparação com soluções humanas.

Referências

- Aamodt, A. and Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59.
- de Freitas, E. P., Zacarias, E., Nunes, R. C., and Silva, L. A. (2024). Interactive multi-level virtual tactical simulation: The development of an autonomy architecture to improve training experience. In *2024 Winter Simulation Conference (WSC)*, pages 2154–2165. IEEE.
- Liu, S., Louis, S. J., and Ballinger, C. A. (2016). Evolving effective microbehaviors in real-time strategy games. *IEEE Transactions on Computational Intelligence and AI in Games*, 8(4):351–362.
- Oh, I.-S., Cho, H., and Kim, K.-J. (2017). Playing real-time strategy games by imitating human players’ micromanagement skills based on spatial analysis. *Expert Systems with Applications*, 67:12–22.
- Pozzer, C. T., Martins, J. B., Fontoura, L. M., Silva, L. A. L., Rutzig, M. B., Nunes, R. C., and de Freitas, E. (2022). Sis-astros: An integrated simulation system for the artillery saturation rocket system (astros). In *Proc. of the 12th Int. Conf. on Simulation and Modeling Methodologies, Technologies and Applications*, pages 194–201.
- Rogers, K. D. and Skabar, A. A. (2014). A micromanagement task allocation system for real-time strategy games. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(1):67–77.