

Implementação Paralela do PSO para o Problema do Roteamento de Veículos Capacitado com uso de OpenMP

Luiz Felipe Xavier Alves¹, Bianca de Almeida Dantas¹

¹Faculdade de Computação – Universidade Federal de Mato Grosso do Sul (UFMS)

Resumo. *O Problema de Roteamento de Veículos Capacitado (PRVC) é um desafio clássico de otimização combinatória com ampla relevância prática. Dada sua natureza NP-difícil, o uso de meta-heurísticas que equilibram diversificação e intensificação, como o Particle Swarm Optimization (PSO) híbrido, é essencial para obter soluções subótimas em tempo hábil. Entretanto, a incorporação de algoritmos robustos de busca local impõe um severo gargalo computacional, limitando a escalabilidade. Neste contexto, este artigo propõe uma estratégia de paralelismo de granulação grossa baseada em OpenMP. Os experimentos conduzidos demonstram a eficácia da abordagem, alcançando expressivos valores de speedup, assegurando a viabilidade temporal para instâncias complexas.*

Abstract. *The Capacitated Vehicle Routing Problem (CVRP) is a classic combinatorial optimization challenge with broad practical relevance. Given its NP-hard nature, the use of metaheuristics that balance exploration and exploitation of the search space, such as the hybrid Particle Swarm Optimization (PSO), is essential to obtain high-quality solutions within a reasonable time. However, the incorporation of robust local search algorithms imposes a severe computational bottleneck, limiting scalability. In this context, this article proposes a coarse-grained parallelism strategy based on OpenMP. The conducted experiments demonstrate the effectiveness of the approach, achieving significant speedup values and ensuring the necessary temporal viability for complex instances.*

Palavras-chave: Problema de Roteamento de Veículos, Particle Swarm Optimization, Computação Paralela, OpenMP

1. Introdução

O Problema do Roteamento de Veículos (PRV), introduzido originalmente por Dantzig e Ramser [Dantzig 1959], desempenha um papel fundamental no desenvolvimento de modelos matemáticos e em aplicações práticas voltadas à logística de transporte e distribuição. Em sua concepção clássica, o problema busca determinar um conjunto otimizado de rotas para que uma frota de veículos atenda às demandas específicas de diversos clientes, sendo um pilar central para a eficiência operacional na movimentação de pessoas e insumos. Dentre suas diversas extensões, destaca-se o PRV Capacitado (PRVC), em que uma frota de veículos com a mesma capacidade deve ser roteada para atender a um conjunto de clientes com demandas conhecidas.

Devido à complexidade computacional inerente ao PRV e suas variantes, classicamente categorizados como NP-difíceis, a obtenção de soluções exatas em tempo polinomial torna-se impraticável para instâncias de grande porte. Portanto, a adoção de meta-heurísticas para a obtenção de soluções subótimas de alta qualidade, dentro de um

tempo computacional viável, consolidou-se como uma abordagem amplamente investigada na literatura. A eficácia desses métodos estocásticos baseia-se no gerenciamento e no equilíbrio entre os mecanismos de diversificação (*exploration*) e intensificação (*exploitation*) do espaço de busca.

A Otimização por Enxame de Partículas (*Particle Swarm Optimization* - PSO), proposta por Kennedy e Eberhart [Kennedy 1995], é uma meta-heurística populacional na qual um conjunto de partículas explora iterativamente o espaço de busca. Durante o processo, a trajetória de cada partícula é atualizada estocasticamente, sendo atraída tanto para a melhor posição histórica descoberta por ela mesma quanto para a melhor solução encontrada pelo enxame. No contexto apresentado, o mecanismo do PSO atua primordialmente na diversificação do espaço de busca enquanto os operadores de busca local, como o 2-opt e o *swap** [Vidal 2021], são empregados como mecanismos de intensificação, encarregados do refinamento das rotas descobertas pelo enxame.

A integração de algoritmos de intensificação proporciona a obtenção de soluções de qualidade significativamente superior. Entretanto, um ônus considerável é imposto ao tempo de execução global do método. Diante desse gargalo de desempenho, torna-se imperativa a otimização computacional do algoritmo, notadamente por meio de técnicas de programação paralela. Nesse cenário, o presente estudo propõe e avalia uma estratégia inicial de paralelismo aplicadas ao PSO hibridizado, com o intuito de mitigar o custo computacional das fases de busca local e viabilizar a resolução de instâncias complexas do PRVC em tempos de processamento competitivos.

2. Metodologia

O presente trabalho concentrou-se na paralelização do algoritmo PSO por meio de diretivas OpenMP, visando obter ganhos de desempenho com baixa complexidade de implementação. A análise de perfilamento de código (*profiling*) evidenciou que o operador *swap** constitui o principal gargalo de desempenho do algoritmo, concentrando cerca de 80% do tempo de CPU. Consequentemente, os esforços de otimização computacional foram direcionados prioritariamente para essa rotina.

Observou-se que um paralelismo de granulação fina limitava a obtenção de *speedup* devido ao *overhead* de gerenciamento das *threads*. Por conseguinte, optou-se por uma abordagem de granulação grossa, inserindo a diretiva de paralelização no laço externo responsável pela aplicação da busca local. Dessa forma, as partículas selecionadas são processadas concorrentemente, o que acelera o processo sem perda na qualidade das soluções obtidas, visto que os algoritmos de intensificação agem de forma independente em cada partícula.

O ambiente experimental foi configurado com 500 iterações e um enxame de 250 partículas, das quais 200 são submetidas às otimizações a cada iteração. Essa parametrização foi empiricamente definida para maximizar a qualidade das soluções do PRVC e, simultaneamente, assegurar uma carga de trabalho densa o suficiente para cada *thread* da CPU, justificando a paralelização.

3. Resultados obtidos

A Tabela 1 sumariza os resultados de 30 execuções independentes do algoritmo PSO proposto, aplicadas ao conjunto de instâncias CMT [Christofides and Eilon 1969], dis-

poníveis no repositório CVRPLib [CVRPLib 2025]. A primeira linha explicita o número de clientes definido pela instância. As duas linhas seguintes apresentam, respectivamente, os tempos computacionais médios (em segundos), acompanhados do desvio padrão, da versão sequencial e da versão paralelizada via OpenMP com 16 *threads*. Por fim, a última linha exibe o *speedup* alcançado, calculado pela razão entre o tempo médio sequencial e o tempo paralelo ($S = T_s/T_p$). Os experimentos foram conduzidos em uma máquina equipada com processador Intel Xeon E5-2620 v4 @ 3.0 GHz e 16 GB de memória RAM DDR4, operando sob o sistema Arch Linux. A compilação do código-fonte utilizou o g++ (versão 16.1.1) acompanhado da *flag* de otimização `-O3`.

Tabela 1. Resultados da execução do PSO (Média $\pm$ Desvio Padrão)

Instância	CMT1	CMT2	CMT3	CMT4	CMT5	CMT11	CMT12
Qt. Clientes	50	75	100	150	199	120	100
Sequencial	14,8 $\pm$ 1,94	32,8 $\pm$ 3,64	65,8 $\pm$ 10,31	139,9 $\pm$ 14,74	257,4 $\pm$ 39,18	138,9 $\pm$ 8,88	60,7 $\pm$ 8,71
OpenMP	4,1 $\pm$ 0,31	8,2 $\pm$ 1,08	14,2 $\pm$ 2,71	26,8 $\pm$ 2,66	51,4 $\pm$ 9,92	24,8 $\pm$ 2,27	11,9 $\pm$ 1,45
<i>Speedup</i>	3,57	4,00	4,64	5,21	5,00	5,61	5,11

Os resultados evidenciam que, apesar da baixa complexidade de implementação da estratégia adotada, obtiveram-se valores de *speedup* expressivos, atingindo um pico de 5,61 na instância CMT11. Observa-se uma clara tendência de melhor aproveitamento da arquitetura paralela à medida que a dimensão do problema cresce: enquanto a instância CMT1 obteve uma aceleração de 3,57, instâncias de maior porte, como a CMT4 e a CMT5, registraram ganhos superiores a 5 vezes. Essa acentuada redução no tempo de execução da fase de busca local viabiliza a incorporação de operadores de intensificação adicionais ou mais robustos, propiciando a convergência para soluções de maior qualidade sem comprometer a viabilidade temporal do algoritmo.

4. Considerações finais

A incorporação da estratégia de paralelismo de granulação grossa ao algoritmo PSO demonstrou-se altamente eficiente na redução do tempo computacional do PRVC. Os expressivos ganhos de *speedup* obtidos validam a viabilidade da abordagem e motivam o aprofundamento nesta linha de pesquisa. Como trabalhos futuros, planeja-se a integração de mecanismos de busca local complementares, visando maximizar a qualidade das soluções, bem como a exploração do paralelismo massivo em aceleradores gráficos (GPGPUs).

Referências

- [Christofides and Eilon 1969] Christofides, N. and Eilon, S. (1969). An algorithm for the vehicle-dispatching problem. *Journal of the Operational Research Society*, 20(3):309–318.
- [CVRPLib 2025] CVRPLib (2025). All instances. <https://galgos.inf.puc-rio.br/cvrplib/en/instances>. Acesso em: 09 ago. 2026.
- [Dantzig 1959] Dantzig, G. B. Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91.
- [Kennedy 1995] Kennedy, J., E. R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948.
- [Vidal 2021] Vidal, T. (2021). Hybrid genetic search for the cvrp: Open-source implementation and swap* neighborhood. *Computers & Operations Research*.