PCC170 - Projeto e Análise de Experimentos Computacionais

Marco Antonio M. Carvalho

Departamento de Computação
Instituto de Ciências Exatas e Biológicas
Universidade Federal de Ouro Preto

UFOP
Universidade Federal de Ouro Preto

decom
departamento de computação

Conteúdo

Aviso

Este material é baseado em minha experiência em pesquisa.

Particularmente, os textos apresentados consistem em sugestões aos alunos e refletem tão somente minha visão pessoal.

Licença

Diretrizes para o relato de experimentos

Linguagem escrita

Atente para a linguagem utilizada.

Não use frases ou parágrafos longos.

Escreva frases curtas e simples: Uma frase com muitas vírgulas deve ser substituída por várias frases separadas por ponto.

Em português, mantenha o tom formal e impessoal.

Diretrizes para o relato de experimentos

Linguagem escrita

Utilize verbos na voz ativa, e.g., sabe-se, entende-se, recomenda-se, e frases em ordem direta, i.e., sujeito + verbo + complementos e/ou adjuntos.

Não utilize superlativos, e.g., “muito”, “o mais”, “extremamente”, “-íssimo”, “maior”, “bastante”, e use adjetivos e advérbios com cautela.

Assegure-se que seu texto foi revisado (*proofread*) por um falante nativo da língua em que foi escrito. Se possível, alguém da área do estudo.

Diretrizes para o relato de experimentos

Organização

Vamos considerar um texto dividido em capítulos ou seções da seguinte maneira:

1. Introdução
2. Descrição do problema
3. Trabalhos relacionados
4. Métodos
5. **Experimentos computacionais**
6. Conclusões e trabalhos futuros

Diretrizes para o relato de experimentos

Organização

A seção de experimentos computacionais pode ainda ser organizada da seguinte maneira:

5.0 Texto introdutório da seção/capítulo;
5.1 Instâncias;
5.2 Experimentos preliminares;
5.3 Limitantes inferiores e/ou superiores;
5.4 Comparação com o estado da arte;
5.5 Análises adicionais.

Diretrizes para o relato de experimentos

Texto introdutório

O primeiro parágrafo dos experimentos computacionais deve conter:

- Indicação dos objetivos do experimento;
- Indicação de disponibilização de resultados detalhados como material suplementar;
- Descrição do ambiente computacional utilizado;
- Descrição do ambiente de codificação e compilação.

5. Computational experiments

A set of computational experiments were performed to tune the hybrid BRKGA components and compare it fairly with the state-of-the-art methods. The next sections describe the benchmark instances considered and report the experimental results regarding parameter tuning and comparison with the state-of-the-art methods. Detailed results for each experiment can be found in the supplementary material.

The computational environment adopted for the computational experiments consisted of a computer with an Intel Xeon E5-2660 2.2 GHz processor with 384 GB of RAM under the CentOS 6.8 operating system and an individual thread score of 1.499 (Passmark, 2018). The BRKGA was implemented in C++, and compiled using GCC 4.8.4 and options -O3 and *-march* = *native*. The OpenMP (2019) shared-memory parallel programming API version 5.0 was used to parallelize the decoding step.

Soares & Carvalho. *Biased random-key genetic algorithm for scheduling identical parallel machines with tooling constraints*. European Journal Of Operational Research, 2020.

4. Computational experiments

A series of computational experiments were performed to tune the ILS parameters, to compare the obtained results by ILS and the exact method with the state-of-the-art method and to measure the quality of the best known solutions against the optimal solutions or new upper bounds reported by the exact method. The next sections describe the computational environments used, the benchmark instances considered and report the experimental results. Detailed results for each experiment can be found in the supplementary material.

Three distinct computational environments were employed in this study.For the constructive heuristic and ILS, the computational environment adopted for the computational experiments consisted of an Intel Core i7-8700 3.20 GHz with 16 GB of RAM under the Ubuntu 18.04 operating system. These methods were implemented using the C++ language and compiled using g++ 4.4.1 and the -O3 optimization option.

The CBM-TSP reduction was coded in C++ and run on an Intel Core i7-4790 3.60 GHz four-core processor with 16 GB RAM. The Concorde solver used was implemented by Hans Mittelmann and made available at the NEOS Server for Optimization (Mittelmann, (2016) using the CPLEX linear programming solver.

fonte: Soares et al. *Heuristic methods to Consecutive Block Minimization*. Computers & Operations Research, 2020.

5. Computational study

All experiments were conducted on a computer with an Intel Xeon E5-2660 processor at 2.6 GHz, with 164 GB of RAM running Ubuntu 18.04 LTS. ILS-LR2L was implemented in C++ and compiled using gcc 7.4.0 and option -O3. All instances, solutions and tables with results are available in a public repository, i.e., Mendeley Data (https://doi.org/10.17632/hkxchx5sxp.1).

fonte: Santos et al. *The impact of loading restrictions on the two-echelon location routing problem*. Computers & Industrial Engineering, v.160, p.107609, 2021.

5. Computational experiments

A set of computational experiments was performed to assess the quality of the proposed hybrid BRKGA and analyze its performance. The following sections describe the instances considered and discuss the experimental results. The detailed results of these experiments are provided in the Supplementary Material. The computational environment of the computational experiments included a computer with Intel Xeon E5-2660 2.2 GHz processor, 384 GB RAM, and CentOS 6.8 operating system. The hybrid BRKGA was implemented in C++ and compiled using GCC 10.1.0 with options -O3 and -march = native. The OpenMP (OpenMP, 2019) shared-memory parallel programming API version 5.0 was used to parallelize the decoding step.

fonte: Soares & Carvalho. *Application of a hybrid evolutionary algorithm to resource-constrained parallel machine scheduling with setup times*. Computers & Operations Research. 2021.

Diretrizes para o relato de experimentos

Instâncias

A descrição das instâncias deve conter:

- Indicação da origem das instâncias;
- Indicação da natureza das instâncias (artificiais ou reais);
- Descrição das características das instâncias;
- Instâncias geradas aleatoriamente devem ser justificadas e esmiuçadas, incluindo a geração de números aleatórios.

Diretrizes para o relato de experimentos

The 45 artificial instances and 5 real-world instances used in the experiments of Haddadi et al. (2015) are considered. Each column of the artificial instances contains at least one nonzero element, and each row contains at least two 1-blocks. As listed in Table 1, the artificial instances are divided into nine groups (*A*–*I*) of five instances each. The real-world instances, whose dimensions are listed in Table 2, arise from a station location problem posed by a German railway company.

fonte: Soares et al. *Heuristic methods to Consecutive Block Minimization*. Computers & Operations Research, 2020.

Table 1
Artificial instances.

Group	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*
Density (%)	2	5	10	2	5	10	2	5	10
Dimensions	100 × 200			100 × 500			100 × 1000		

fonte: Soares et al. *Heuristic methods to Consecutive Block Minimization*. Computers & Operations Research, 2020.

5.1. New instance set

To run experiments with ILS-LR2L, and in order to stimulate further research regarding the 2E-LRP2L, instances were generated based on real-world data. A transportation company of large equipment provided locations of facilities (platforms and satellites) and customers, vehicle and items sizes (l_v, w_v and l_m, w_m), and approximately two months worth of customer demands and their respective day of delivery. There are up to four platforms and eight satellites which can be operational depending on the total customer demand. Vehicles have equal width (2.5 m) but three different lengths (17.6, 8.5 and 7 m), while items are at most 5 meters long and 2.5 meters wide.

A travel time matrix provides realistic travel times between all pairs of locations. Meanwhile, service time st_m is based on item m's dimensions and may range from 5 to 20 min. Requests were divided into 32 instances according to their delivery date and range from 81 to 274 customer requests. Due to privacy issues, all identifying features such as names and physical locations have been omitted.

fonte: Santos et al. *The impact of loading restrictions on the two-echelon location routing problem*. Computers & Industrial Engineering, v.160, p.107609, 2021.

The cost and capacity of facilities and the cost associated with vehicles were not provided by the company and were generated for this paper in accordance with the 2E-LRP benchmark instances. We first analyzed how the cost is distributed among the many components present in a 2E-LRP solution. Next, costs were calibrated to generate 2E-LRP2L instances with similar proportions. The cost breakdown is illustrated in Appendix C (Fig. C.11), where the total cost of an initial solution is divided in percentages considering first and second echelons, the cost of open facilities and routes, as well as traveling time and vehicle cost. The reported percentages are an average across all instances.

Tuning facility capacities correctly is important in order to create instances that represent the range of different sizes observed in practice. The idea is to assign capacities to facilities in such a way that all customers may be served without needing to open all facilities. For each instance, given a set of customers and their respective demand, the total area of all requested items was taken into account when generating the capacity of facilities. Satellite capacities were randomly generated between 17% and 30% of total item area. In this way, solutions must open at least four satellites out of the eight available. Platform capacities were selected between 40% and 70% of total item area, which results in solutions with two or three open platforms out of four.

fonte: Santos et al. *The impact of loading restrictions on the two-echelon location routing problem*. Computers & Industrial Engineering, v.160, p.107609, 2021.

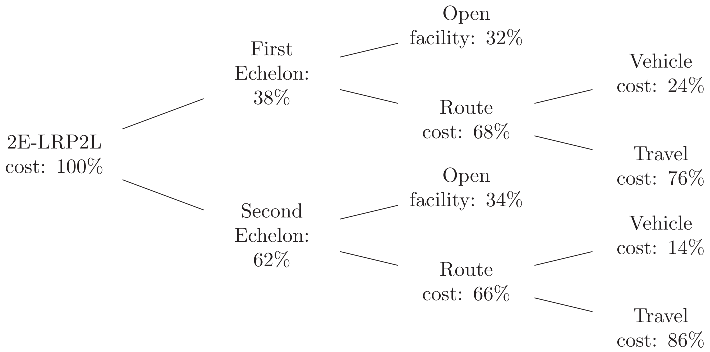

Fig. C.11. The average cost breakdown of 2E-LRP2L solutions.

fonte: Santos et al. *The impact of loading restrictions on the two-echelon location routing problem*. Computers & Industrial Engineering, v.160, p.107609, 2021.

5.1. Instances

The 2880 benchmark instances provided by (Beezão et al., 2017) are considered. The instances are separated into two sets, IPMTC-I and IPMTC-II, each containing 1440 instances. Table 1 presents the characteristics of each set. The column headings represent the numbers of instances (#), machines (m), jobs (n), and tools (l).

fonte: Soares & Carvalho. *Application of a hybrid evolutionary algorithm to resource-constrained parallel machine scheduling with setup times*. Computers & Operations Research. 2021.

Table 1
Characteristics of the instances.

set	#	m	n	l
IPMTC-I	240	2	{8, 15, 25}	{15, 20}
	480	3	{15, 25}	{15, 20}
	720	4	{25}	{15, 20}
IPMTC-II	360	{3, 4, 5}	50	{30, 40}
	480	{4, 5, 6, 7}	100	{30, 40}
	600	{6, 7,8, 9, 10}	200	{30, 40}

fonte: Soares & Carvalho. *Application of a hybrid evolutionary algorithm to resource-constrained parallel machine scheduling with setup times*. Computers & Operations Research. 2021.

Diretrizes para o relato de experimentos

Experimentos preliminares

Os experimentos preliminares incluem todos os testes realizados antes da comparação com o estado da arte:

- Definição de parâmetros, incluindo tabela de valores considerados e selecionados;
- Descrição do que está sendo avaliado nos experimentos;
- Resultados resumidos;
- Análise e conclusão sobre os resultados.

Diretrizes para o relato de experimentos

To define the parameters, we employed the *irace* offline method for automatic configuration of optimization algorithms (López-Ibáñez et al., 2016). Given a set of instances for an optimization problem and a set of candidate values, the irace determines the best combination of values for the parameters. In this experiment, the training set is a representative subset of the set of instances, consisting of 5 instances (10% of the total) were randomly selected among all subgroups of the instances. In preliminary experiments we considered different values for the parameters. The ILS was not sensitive to small changes in the parameter values, then, the values were categorized, omitting intervals and narrowed down the options to the values presented in Table 3, which also presents the values selected by irace.

fonte: Soares et al. *Heuristic methods to Consecutive Block Minimization*. Computers & Operations Research, 2020.

Diretrizes para o relato de experimentos

Table 3
Parameter tuning.

Parameter	Options	Selected Value
α (%)	{5, 10, 15, 20, 25, 30, 35, 40, 45, 50}	10
max_iter	{50, 100, 150, 200, 250}	150
max_time	$\{\frac{n}{3}, \frac{n}{2}, n, 2 \times n\}$	$\frac{n}{2}$

fonte: Soares et al. *Heuristic methods to Consecutive Block Minimization*. Computers & Operations Research, 2020.

5.2. Preliminary experiments

The ALNS implementation requires a few parameters to be defined beforehand. For the experiments in this study, these parameters were determined using the *irace* (López-Ibáñez, Dubois-Lacoste, Cáceres, Birattari, & Stützle, 2016) tool, a package implemented in the R language. Given a set of instances and a set of possible values for each of the parameters, the irace package determines an appropriate combination of values for the parameters. For this test, 25 instances were randomly selected from the HB set along with 10% of the instances from the Small, Grid, and Rome Graphs sets. Table 1 shows the range of the values that irace considered when defining each parameter as well as the defined values.

fonte: Santos & Carvalho. *Tailored heuristics in adaptive large neighborhood search applied to the cutwidth minimization problem*. European Journal Of Operational Research, 2021.

Table 1
Values considered by irace for each parameter and respective chosen values.

Parameters	Range	Defined values
σ_1	{10, 15, 25, 30, 35, 40, 45, 50}	50
σ_2	{10, 15, 25, 30, 35, 40, 45, 50}	15
σ_3	{10, 15, 25, 30, 35, 40, 45, 50}	25
Reaction factor ρ	[0.00.. 1.00]	0.85
Initial worsening percentage p_s	[0.50.. 1.00]	0.85
Final worsening percentage p_e	[0.00.. 0.50]	0.45
Number of iterations	{600, 1000, 1300, 1600, 2000, 2300, 2600, 3000}	3000
Segments θ	{50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300}	200
Noise percentage *noiseAcc*	{0.02, 0.05, 0.07, 0.1}	0.07
Noise percentage *noiseIn*	{0.02, 0.05, 0.07, 0.1}	0.07

fonte: Santos & Carvalho. *Tailored heuristics in adaptive large neighborhood search applied to the cutwidth minimization problem*. European Journal Of Operational Research, 2021.

Diretrizes para o relato de experimentos

5.5. Preliminary experiment

A preliminary experiment was conducted in order to analyze the individual BRKGA components against the hybrid version. This experiment considered the hybrid BRKGA, the original BRKGA without hybridization and the VND alone with random initial solutions. Owing to the randomness present in the methods, we considered ten independent runs and only the best solutions achieved by each method were considered in the analyses.

Considering the hybrid BRKGA as the reference and the RCPMS I set, the original BRKGA reported an average percentage distance of 5.01%, ranging between 0.00% and 28.70%. The VND reported an average percentage distance of 26.04%, ranging between 0.00% and 131.62%. For the RCPMS II set, the BRKGA reported an average percentage distance of 67.94%, ranging between 0.68% and 159.85%. The VND reported an average percentage distance of 126.28%, ranging between 5.87% and 244.54%. Both original BRKGA and VND alone presented solution values worse than the upper bound values, hence, these methods are not considered in the next experiment. The detailed results of this experiment are provided in the Supplementary Material.

fonte: Soares & Carvalho. *Application of a hybrid evolutionary algorithm to resource-constrained parallel machine scheduling with setup times*. Computers & Operations Research. 2021.

Leitura recomendada

- Soares, Leonardo C.R.; Carvalho, Marco A.M. *Application of a hybrid evolutionary algorithm to resource-constrained parallel machine scheduling with setup times*. Computers & Operations Research. 2021.
- Gandra, V. M. S.; Carvalho, M. A. M. *Tailored heuristics in adaptive large neighborhood search applied to the cutwidth minimization problem*. European Journal Of Operational Research, 2021.
- Santos Gandra, Vinícius Martins; Çalık, Hatice; Wauters, Tony; Toffolo, Túlio A.M.; Moreira De Carvalho, Marco Antonio; Berghe, Greet Vanden *The impact of loading restrictions on the two-echelon location routing problem*. Computers & Industrial Engineering, v.160, p.107609, 2021.

Leitura recomendada

- Soares, Leonardo C.R.; Carvalho, Marco A.M. *Biased random-key genetic algorithm for scheduling identical parallel machines with tooling constraints*. European Journal Of Operational Research, 2020.
- Soares, Leonardo C.R.; Reinsma, Jordi Alves; Nascimento, Luis H.L.; Carvalho, Marco A.M. *Heuristic methods to Consecutive Block Minimization*. Computers & Operations Research, 2020.

Dúvidas?