

## Bibliografia

- [1] ALVIM, A. C. F.. **Resultados Computacionais Detalhados para BP e PCmax**. Disponíveis em <http://www.inf.puc-rio.br/~alvim/adriana/tese.html>.
- [2] ALVIM, A. C. F.; GLOVER, F.; RIBEIRO, C. C. ; ALOISE, D. J.. **Local search for the bin packing problem**. In: EXTENDED ABSTRACTS OF THE 3RD METAHEURISTICS INTERNATIONAL CONFERENCE, p. 7–12, Angra dos Reis, 1999.
- [3] ALVIM, A. C. F.; GLOVER, F.; RIBEIRO, C. C. ; ALOISE, D. J.. **A hybrid improvement heuristic for the bin packing problem**. In: EXTENDED ABSTRACTS OF THE 4RD METAHEURISTICS INTERNATIONAL CONFERENCE, p. 63–68, Porto, 2001.
- [4] ARGÜELLO, M. F.; FEO, T. A. ; GOLDSCHMIDT, O.. **Randomized methods for the number partitioning problem**. Computers and Operations Research, 23:103–111, 1996.
- [5] BARR, R.; GOLDEN, B.; KELLY, J.; RESENDE, M. ; STEWART, W.. **Designing and reporting on computational experiments with heuristic methods**. Journal of Heuristics, 1:9–32, 1995.
- [6] BEASLEY, J. E.. **OR-library: Distributing test problems by electronic mail**. Journal of the Operational Research Society, 41:1069–1072, 1990.
- [7] BLACKSTONE, JR., J. H.; PHILLIPS, D. T.. **An improved heuristic for minimizing makespan among  $m$  identical parallel processors**. Computers and Industrial Engineering, 5:279–287, 1981.
- [8] BŁAŻEWICZ, J.. **Selected topics in scheduling theory**. In: Laporte, G.; Martello, S.; Minoux, M. ; Ribeiro, C., editores, SURVEYS IN COMBINATORIAL OPTIMIZATION, volume 31 de **Annals of Discrete Mathematics**, p. 1–60. North-Holland, 1987.

- [9] BROWN, A. R.. **Optimum Packing and Depletion**. Macdonald & Co. Ltd, London, 1971.
- [10] BRUNO, J. L.; COFFMAN JR., E. G. ; SETHI, R.. **Scheduling independent tasks to reduce mean finishing time**. Communications of the Association for Computing Machinery, 17:382–387, 1974.
- [11] CHENG, T.; SIN, C.. **A state-of-the-art review of parallel-machine scheduling research**. European Journal of Operational Research, 47:271–292, 1990.
- [12] COFFMAN, JR., E. G.; GALAMBOS, G.; MARTELLO, S. ; VIGO, D.. **Bin packing approximation algorithms: Combinatorial analysis**. In: Du, D. Z.; Pardalos, P. M., editores, HANDBOOK OF COMBINATORIAL OPTIMIZATION, p. 151–207. Kluwer Academic Publishers, 1999.
- [13] COFFMAN JR., E. G.; GAREY, M. R. ; JOHNSON, D. S.. **An application of bin-packing to multiprocessor scheduling**. SIAM Journal on Computing, 7:1–17, 1978.
- [14] COFFMAN, JR., E. G.; GAREY, M. R. ; JOHNSON, D. S.. **Approximation algorithms for bin-packing: A survey**. In: Hochbaum, D., editor, APPROXIMATION ALGORITHMS FOR NP-HARD PROBLEMS, p. 46–93. PWS Publishing, 1997.
- [15] COSTA, A. M.; VARGAS, P. A.; ZUBEN, F. J. V. ; FRANÇA, P. M.. **Makespan minimization on parallel processors: An immune-based approach**. In: PROCEEDINGS OF THE 2002 CONGRESS ON EVOLUTIONARY COMPUTATION, p. 920–925, Honolulu, 2002.
- [16] DELL’AMICO, M.; MARTELLO, S.. **Optimal scheduling of tasks on identical parallel processors**. ORSA Journal on Computing, 7:191–200, 1995.
- [17] DELL’OLMO, P.; KELLERER, H.; SPERANZA, M. ; TUZA, Z.. **A 13/12 approximation algorithm for bin packing with extendable bins**. Information Processing Letters, 65:229–233, 1998.
- [18] DYCKHOFF, H.. **A typology of cutting and packing problems**. European Journal of Operational Research, 44:145–159, 1990.
- [19] EILON, S.; CHRISTOFIDES, N.. **The loading problem**. Management Science, 17:259–267, 1971.

- [20] FALKENAUER, E.. **A hybrid grouping genetic algorithm for bin packing.** *Journal of Heuristics*, 2:5–30, 1996.
- [21] FATEMI-GHOMI, S.; JOLAI-GHAZVINI, F.. **A pairwise interchange algorithm for parallel machine scheduling.** *Production Planning and Control*, 9:685–689, 1998.
- [22] FEKETE, S. P.; SCHEPERS, J.. **New classes of fast lower bounds for bin packing problems.** *Mathematical Programming*, 91:11–31, 2001.
- [23] FERNANDEZ DE LA VEGA, W.; LUEKER, G. S.. **Bin packing can be solved within  $1 + \epsilon$  in linear time.** *Combinatorica*, 1:349–355, 1981.
- [24] FINN, G.; HOROWITZ, E.. **A linear time approximation algorithm for multiprocessor scheduling.** *BIT*, 19:312–320, 1979.
- [25] FLESZAR, K.; HINDI, K. S.. **New heuristics for one-dimensional bin packing.** *Computers and Operations Research*, 29:821–839, 2002.
- [26] FRANÇA, P. M.; GENDREAU, M.; LAPORTE, G. ; MÜLLER, F. M.. **A composite heuristic for the identical parallel machine scheduling problem with minimum makespan objective.** *Computers and Operations Research*, 21:205–210, 1994.
- [27] FRANGIONI, A.; NECCIARI, E. ; SCUTELLÀ, M. G.. **A multi-exchange neighborhood for minimum makespan machine scheduling problems.** Relatório Técnico TR-00-17, Dipartimento di Informatica, Università di Pisa, Pisa, 2000.
- [28] FRANGIONI, A.; SCUTELLÀ, M. G. ; NECCIARI, E.. **Multi-exchange algorithms for the minimum makespan machine scheduling problem.** Relatório Técnico TR-99-22, Dipartimento di Informatica, Università di Pisa, Pisa, 1999.
- [29] FRIESEN, D. K.. **Tighter bounds for the multifit processor scheduling algorithm.** *SIAM Journal on Computing*, 13:170–181, 1984.
- [30] FRIESEN, D. K.; LANGSTON, M. A.. **Evaluation of a multifit-based scheduling algorithm.** *Journal of Algorithms*, 7:35–59, 1986.

- [31] FRIESEN, D. K.; LANGSTON, M. A.. **Analysis of a compound bin packing algorithm**. SIAM Journal on Discrete Mathematics, 4:61–79, 1991.
- [32] GAREY, M. R.; JOHNSON, D. S.. **Computers and Intractability: A Guide to the Theory of NP-Completeness**. W.H. Freeman and Company, New York, 1979.
- [33] GENT, I. P.. **Heuristic solution of open bin packing problems**. Journal of Heuristics, 3:299–304, 1998.
- [34] GENT, I. P.; WALSH, T.. **Analysis of heuristics for number partitioning**. Computational Intelligence, 14:430–451, 1998.
- [35] GILMORE, P.; GOMORY, R.. **A linear programming approach to the cutting-stock problem**. Operations Research, 9:849–859, 1961.
- [36] GILMORE, P.; GOMORY, R.. **A linear programming approach to the cutting-stock problem II**. Operations Research, 11:863–888, 1963.
- [37] GLOVER, F.; LAGUNA, M.. **Tabu Search**. Kluwer Academic Publishers, Boston, 1997.
- [38] GRAHAM, R. L.. **Bounds for certain multiprocessing anomalies**. The Bell System Technical Journal, 45:1563–1581, 1966.
- [39] GRAHAM, R. L.. **Bounds on multiprocessing timing anomalies**. SIAM Journal of Applied Mathematics, 17:416–429, 1969.
- [40] GRAHAM, R. L.. **Combinatorial scheduling theory**. In: Steen, L. A., editor, MATHEMATICS TODAY: TWELVE INFORMAL ESSAYS, p. 183–211. Springer-Verlag, 1978.
- [41] GRAHAM, R. L.; LAWLER, E. L.; LENSTRA, J. K. ; RINNOOY KAN, A. H. G.. **Optimization and approximation in deterministic sequencing and scheduling: A survey**. Annals of Discrete Mathematics, 5:287–326, 1979.
- [42] GRUPO DE OTIMIZAÇÃO DA UNIVERSIDADE FEDERAL DE SANTA MARIA (UFSM). Disponível em <http://glover.ce.ufsm.br/>. Último acesso em 13 de abril de 2003.

- [43] GUPTA, J. N. D.; HO, J. C.. **A new heuristic algorithm for the one-dimensional bin-packing problem.** *Production Planning and Control*, 10:598–603, 1999.
- [44] HANSEN, P.; MLADENOVIĆ, N.. **An introduction to variable neighborhood search.** In: Voss, S.; Martello, S.; Osman, I. H. ; Roucairol, C., editores, *META-HEURISTICS, ADVANCES AND TRENDS IN LOCAL SEARCH PARADIGMS FOR OPTIMIZATION*, p. 433–458. Kluwer Academic Publishers, 1998.
- [45] HO, J. C.; WONG, J. S.. **Makespan minimization for  $m$  parallel identical processors.** *Naval Research Logistics*, 42:935–948, 1995.
- [46] HOCHBAUM, D. S.; SHAMIR, R.. **Strongly polynomial algorithms for the high multiplicity scheduling problems.** *Operations Research*, 39:648–653, 1991.
- [47] HOCHBAUM, D. S.; SHMOYS, D.. **A packing problem you can almost solve by sitting on your suitcase.** *SIAM Journal on Algebraic and Discrete Methods*, 7:247–257, 1986.
- [48] HOCHBAUM, D. S.; SHMOYS, D. B.. **Using dual approximation algorithms for scheduling problems: Theoretical and practical results.** *Journal of ACM*, 34:144–162, 1987.
- [49] HÜBSCHER, R.; GLOVER, F.. **Applying tabu search with influential diversification to multiprocessor scheduling.** *Computers and Operations Research*, 21:877–884, 1994.
- [50] HUNG, M.; BROWN, J.. **An algorithm for a class of loading problems.** *Naval Research Logistics Quarterly*, 17:289–297, 1978.
- [51] JOHNSON, D.. **Fast algorithms for bin packing.** *Journal of Computer and System Sciences*, 8:272–314, 1974.
- [52] JOHNSON, D.; DEMERS, A.; ULLMAN, J. D.; GAREY, M. R. ; GRAHAM, R. L.. **Worst case performance bounds for simple one-dimensional packing algorithms.** *SIAM Journal on Computing*, 3:299–325, 1974.
- [53] JOHNSON, D. S.; GAREY, M. R.. **A 71/60 theorem for bin packing.** *Journal of Complexity*, 1:65–106, 1985.

- [54] KARMAKAR, N.; KARP, R. M.. **An efficient approximation scheme for the one-dimensional bin-packing problem.** In: PROCEEDINGS OF THE 23RD ANNUAL SYMPOSIUM ON THE FOUNDATIONS OF COMPUTER SCIENCE, p. 312–320, Chicago, 1982. IEEE Computer Society.
- [55] KARMARKAR, N.; KARP, R. M.. **The differencing method of set partitioning.** Relatório Técnico UCB/CSD 82/113, Computer Science Division, University of California, Berkeley, 1982.
- [56] KENYON, C.. **Best-fit bin-packing with random order.** In: PROCEEDINGS OF THE SEVENTH ANNUAL ACM-SIAM SYMPOSIUM ON DISCRETE ALGORITHMS, p. 359–364, 1996.
- [57] KLEIN, R.; SCHOLL, A.. **Bin Packing.** Disponível em <http://www.bwl.tu-darmstadt.de/bwl3/forsch/projekte/binpp/index.htm>. Último acesso em 13 de abril de 2003.
- [58] LABBÉ, M.; LAPORTE, G. ; MARTELLO, S.. **An exact algorithm for the dual bin packing problem.** Operations Research Letters, 17:9–18, 1995.
- [59] LANGSTON, M. A.. **Improved 0/1-interchange scheduling.** BIT, 3:282–290, 1982.
- [60] LAWLER, E. L.; LENSTRA, J. K.; RINNOOY KAN, A. H. G. ; SHMOYS, D. B.. **Sequencing and scheduling: algorithms and complexity.** In: Graves, S. C.; Zipkin, P. H. ; Rinnooy Kan, A., editores, LOGISTICS OF PRODUCTION AND INVENTORY: HANDBOOKS IN OPERATIONS RESEARCH AND MANAGEMENT SCIENCE, volume 4, p. 445–522. North-Holland, Amsterdam, 1993.
- [61] LEE, C. C.; LEE, D. T.. **A simple on-line bin packing algorithm.** Journal of the Association for Computing Machinery, 32:562–572, 1985.
- [62] LEE, C. Y.; MASSEY, D.. **Multiprocessor scheduling: An extension of the MULTIFIT algorithm.** Journal of Manufacturing System, 7:25–32, 1988.
- [63] LEE, C. Y.; MASSEY, D.. **Multiprocessor scheduling: Combining LPT and MULTIFIT.** Discrete Applied Mathematics, 20:233–242, 1988.

- [64] MARTELLO, S.; TOTH, P.. **Lower bounds and reduction procedures for the bin packing problem**. Discrete Applied Mathematics, 28:59–70, 1990.
- [65] MARTELLO, S.; TOTH, P.. **Knapsack problems: algorithms and computer implementations**. Wiley, London, 1990.
- [66] MCGEOCH, C.. **Toward an experimental method for algorithm simulation**. INFORMS Journal on Computing, 8:1–15, 1996.
- [67] MCNAUGHTON, R.. **Scheduling with deadlines and loss functions**. Management Science, 6:1–12, 1959.
- [68] MOTWANI, R.. **Lecture notes on approximation algorithms: Volume I**. Relatório Técnico CS-TR-921435, Department of Computer Science, Stanford University, Stanford, 1992.
- [69] MUKHACHEVA, E. A.; BELOV, G. N.; KARTACK, V. M. ; MUKHACHEVA, A. S.. **Linear one-dimensional cutting-packing problems: numerical experiments with the sequential value correction method (SVC) and a modified branch-and-bound method (MBB)**. Pesquisa Operacional, 20:153–168, 2000.
- [70] MÜLLER, F. M.. **Algoritmos heurísticos e exatos para resolução do problema de seqüenciamento em processadores paralelos**. Tese de Doutorado, Departamento de Engenharia de Sistemas, Universidade Estadual de Campinas, Campinas, 1993.
- [71] NECCIARI, E.. **Istances of machine scheduling problems**. Desenvolvido pelo Dipartimento di Informatica, Università di Pisa. Disponível em <http://www.di.unipi.it/di/groups/optimize/Data/index.html/>. Último acesso em 21 de novembro de 2001.
- [72] REEVES, C. R.. **Modern heuristic techniques for combinatorial problems**. Wiley, New York, 1993.
- [73] SAHNI, S.. **Algorithms for scheduling independent tasks**. Journal of the Association for Computing Machinery, 23:116–127, 1976.
- [74] SCHEITHAUER, G.; TERNO, J.. **A branch & bound algorithm for solving one dimensional cutting stock problems exactly**. Applicationes Mathematicae, 23:151–167, 1995.

- [75] SCHEITHAUER, G.; TERNO, J.; MÜLLER, A. ; BELOV, G.. **Solving one-dimensional cutting stock problems exactly with a cutting plane algorithm**. Journal of the Operational Research Society, 52:1390–1401, 2001.
- [76] SCHOENFIELD, J. E.. **Fast, exact solution of open bin packing problems without linear programming**. Relatório Técnico, US Army Space & Missile Defense Command, 2002.
- [77] SCHOLL, A.; KLEIN, R. ; JÜRGENS, C.. **BISON: A fast hybrid procedure for exactly solving the one-dimensional bin packing problem**. Computers and Operations Research, 24:627–645, 1997.
- [78] SCHOLL, A.; VOSS., S.. **Simple assembly line balancing - heuristic approaches**. Journal of Heuristics, 2:217–244, 1996.
- [79] SCHRAGE, L.. **A more portable Fortran random number generator**. ACM Transactions on Mathematical Software, 5:132–138, 1979.
- [80] SCHWERIN, P.; WÄSCHER, G.. **The bin-packing problem: A problem generator and some numerical experiments with FFD packing and MTP**. International Transactions in Operational Research, 4:337–389, 1997.
- [81] SCHWERIN, P.; WÄSCHER, G.. **A new lower bound for the bin-packing problem and its integration to MTP**. Pesquisa Operacional, 19:111–129, 1999.
- [82] SICUP - THE SPECIAL INTEREST GROUP ON CUTTING AND PACKING. Disponível em <http://www.apdio.pt/sicup/Sicuphomepage/research.htm>. Último acesso em 13 de abril de 2003.
- [83] SOMA, N. Y.; YANASSE, H. H. ; MACULAN, N.. **A heuristic for the bin-packing problem**. Relatório Técnico, Instituto Tecnológico de Aeronáutica, São José dos Campos, 1999.
- [84] VALÉRIO DE CARVALHO, J. M.. **Exact solution of bin-packing problems using column generation and branch-and-bound**. Annals of Operations Research, 86:629–659, 1999.
- [85] VANCE, P.. **Branch-and-price algorithms for the one-dimensional cutting stock problem**. Computational Optimization and Applications, 9:212–228, 1998.

- [86] VANCE, P.; BARNHART, C.; JOHNSON, E. ; NEMHAUSER, G..  
**Solving binary cutting stock problems by column generation  
and branch-and-bound.** Computational Optimization and Applica-  
tions, 3:111–130, 1994.
- [87] VANDERBECK, F..  
**Computational study of a column genera-  
tion algorithm for bin packing and cutting stock problems.**  
Mathematical Programming, 86:565–594, 1999.
- [88] WÄSCHER, G.; GAU, T..  
**Heuristics for the one-dimensional  
cutting stock problem: A computational study.** OR Spektrum,  
18:131–144, 1996.
- [89] YAO, A..  
**New algorithms for bin packing.** Journal of the ACM,  
27:207–227, 1980.
- [90] YUE, M..  
**On the exact upper bound for the multifit processor  
scheduling algorithm.** Annals of Operations Research, 24:233–259,  
1990.

## A

### Exemplos da aplicação do limite $L_\vartheta$ a instâncias de BP

Nesta seção apresenta-se exemplos da aplicação do Limite  $L_\vartheta$ , apresentado na Seção 2.2.4.6, a instâncias de BP. Estas instâncias são introduzidas na Seção 3.4.2.

**Exemplo A.1** *Considera-se a instância N2W1B1R0 .BPP, do conjunto set\_2 do GRUPO-II, definida por*

$$\begin{aligned}
 n &= 100 \\
 w &= (393, 390, 390, 389, 386, 382, 381, 381, 381, 380, 379, 379, 377, 375, 372, \\
 &\quad 370, 368, 368, 367, 366, 366, 365, 365, 363, 361, 359, 359, 357, 357, 356, \\
 &\quad 355, 355, 355, 353, 352, 352, 347, 347, 346, 344, 344, 341, 337, 336, 334, \\
 &\quad 334, 333, 333, 333, 332, 332, 329, 328, 326, 326, 324, 324, 319, 319, 318, \\
 &\quad 316, 312, 312, 311, 310, 309, 307, 306, 305, 305, 301, 300, 299, 298, 298, \\
 &\quad 296, 296, 294, 292, 290, 289, 289, 286, 284, 284, 283, 281, 280, 278, 278, \\
 &\quad 277, 277, 273, 273, 272, 271, 269, 268, 268, 267) \\
 C &= 1000.
 \end{aligned}$$

*Calcula-se o limite inferior  $L_1 = \lceil 32894/1000 \rceil = 33$  e faz-se  $m = 33$ . Pelo menos uma caixa tem que ter  $\lceil n/m \rceil = \lceil 100/33 \rceil = 4$  objetos. Somando-se os pesos dos quatro objetos mais leves verifica-se que o peso  $267 + 268 + 268 + 269 = 1073$  não cabe em uma caixa, pois a capacidade máxima das caixas é 1000. Desta forma, de acordo com a Proposição 2.4,  $m + 1 = 34$  é um limite inferior válido.*

**Exemplo A.2** *Considera-se a instância N2W1B2R0.BPP, do conjunto set\_2 do GRUPO-II, definida por*

$$n = 100$$

$$w = (494, 493, 490, 488, 477, 474, 470, 465, 462, 449, 449, 448, 447, 447, 444, 442, 436, 436, 432, 428, 428, 423, 421, 418, 417, 416, 410, 409, 408, 405, 402, 401, 401, 400, 399, 395, 395, 394, 388, 387, 387, 380, 378, 378, 372, 372, 364, 364, 360, 356, 354, 347, 346, 346, 332, 331, 331, 326, 317, 317, 315, 314, 313, 312, 308, 305, 303, 301, 299, 295, 294, 292, 291, 288, 288, 283, 282, 279, 278, 275, 272, 270, 268, 268, 255, 255, 242, 240, 237, 236, 234, 215, 211, 208, 206, 206, 203, 196, 191, 167)$$

$$C = 1000.$$

*Calcula-se o limite inferior  $L_1 = \lceil 34841/1000 \rceil = 35$  e faz-se  $m = 35$ . Verifica-se que o número máximo de objetos por caixa é  $\Theta = 5$ . Calcula-se o número mínimo de objetos por caixa e tem-se  $\vartheta = 2$ . Como  $\Theta > \vartheta + 1$ , a Proposição 2.9 não pode ser verificada. Calcula-se então o número mínimo de caixas  $\underline{m}_\vartheta$  com exatamente  $\vartheta$  objetos e tem-se  $\underline{m}_\vartheta = 35 - (100 - 70) = 5$ . Colocando-se os  $\underline{m}_\vartheta \vartheta$  objetos mais pesados em  $\underline{m}_\vartheta$  caixas, sobram os  $n - (\underline{m}_\vartheta \vartheta) = 90$  objetos mais leves para serem colocados em  $m - \underline{m}_\vartheta = 30$  caixas. Somando-se o peso destes objetos tem-se um peso total igual a 30079 que não cabe nas 30 caixas restantes. Desta forma, de acordo com a Proposição 2.8,  $m + 1 = 36$  é um limite inferior válido.*

**Exemplo A.3** *Considera-se a instância HARD0.BPP, do conjunto set\_3 do GRUPO-II, definida por*

$$n = 200$$

$$w = (34978, 34849, 34703, 34608, 34598, 34524, 34356, 34308, 34069, 34049, 33895, 33842, 33806, 33738, 33716, 33590, 33546, 33507, 33468, 33465, 33383, 33190, 33075, 32976, 32897, 32762, 32696, 32638, 32553, 32398, 32230, 32176, 31967, 31954, 31903, 31782, 31724, 31686, 31597, 31561, 31532, 31499, 31346, 30943, 30915, 30869, 30766, 30683, 30678, 30644, 30559, 30448, 30315, 30238, 30125, 29974, 29947, 29890, 29886, 29858, 29856, 29783, 29697, 29438, 29427, 29301, 29174, 29173, 29123, 29117, 29116, 29095, 29094, 29063, 29041, 29038, 28977, 28946, 28921, 28910, 28842, 28703, 28360, 28350, 28305, 28302, 28225, 28160, 28094, 28040, 28020, 27901, 27775, 27765, 27688, 27439, 27425, 27394, 27365, 27349, 27284, 27180, 26935, 26881, 26867, 26795, 26703, 26651, 26550, 26432, 26375, 26368, 26244, 26204, 26192, 26181, 26158, 26133, 26067, 25945, 25906, 25759, 25698, 25688, 25652, 25615, 25530, 25528, 25366, 25324, 25273, 25142, 24852, 24846, 24658, 24592, 24564, 24463, 24457, 24374, 24359, 24332, 23987, 23956, 23952, 23932, 23895, 23837, 23795, 23774, 23663, 23621, 23502, 23453, 23430, 23366, 23178, 23090, 22991, 22942, 22743, 22442, 22432, 22415, 22338, 22134, 22081, 22014, 21950, 21948, 21796, 21784, 21727, 21722, 21557, 21498, 21480, 21315, 21193, 21127, 21060, 20997, 20837, 20813, 20693, 20693, 20686, 20677, 20676, 20664, 20663, 20634, 20616, 20570, 20566, 20496, 20441, 20307, 20226, 20114)$$

$$C = 1000.$$

*Calcula-se o limite inferior  $L_1 = \lceil 5440282/100000 \rceil = 55$  e faz-se  $m = 55$ . Verifica-se que o número máximo de objetos por caixa é  $\Theta = 4$ . Calcula-se o número médio de objetos por caixa  $\lfloor n/m \rfloor = 3$  e o número mínimo de objetos por caixa  $\vartheta = 2$ . Como  $\Theta > \vartheta + 1$ , a Proposição 2.9 não pode ser verificada. Como  $\vartheta < \lfloor n/m \rfloor$  é possível fazer a seguinte redução. Retira-se os  $\vartheta = 2$  objetos mais pesados para tentar provar que uma solução com  $\vartheta = 2$  é inviável. Considera-se a instância com 198 objetos de pesos  $w_{\vartheta+1}, \dots, w_n$  e o limite  $m - 1 = 54$  caixas, calcula-se  $\Theta = 4$ ,  $\lfloor n/m \rfloor = 3$  e  $\vartheta = 3$ . Como  $\Theta = \vartheta + 1$ , calcula-se  $m_\vartheta = 18$ ,  $m_\Theta = 36$  e  $C_\vartheta(54) = 100556$ . Uma vez que  $C_\vartheta(54) > C$ , sabe-se que não existe solução viável com  $\vartheta = 2$ ,  $n = 200$  e  $m = 55$ , logo faz-se  $\vartheta = 3$ . Calcula-se o limite  $L_\vartheta$  para a instância original com  $n = 200$ ,  $\Theta = 4$ ,  $\vartheta = 3$  e o limite no número de caixas igual a  $m = 55$ . Tem-se que  $\Theta = \vartheta + 1$  e calcula-se  $m_\vartheta = 20$ ,  $m_\Theta = 35$  e*

$C_{\mathfrak{g}}(55) = 100012$ . Como  $C_{\mathfrak{g}}(55) > C$ , de acordo com a Proposição 2.9 tem-se que  $m + 1 = 56$  é um limite inferior válido.

## B

### Algoritmo 1/5-dual-BP para BP

A seguir apresenta-se o pseudo-código do algoritmo 1/5-dual-BP [48] para o problema BP (Figura B.1), abordado na Seção 4.2.1. Escala-se os pesos  $w_i$  dos  $n$  objetos ( $w_i/C$ ) para pertencerem ao intervalo  $[0, 1]$  e a capacidade da caixa  $C$  assume o valor igual a 1. O objetivo é encontrar um empacotamento dos  $n$  objetos onde a soma dos pesos em cada caixa não exceda  $6/5$ . A notação  $L[u_1, \dots, u_k]$  será utilizada para denotar o conjunto de  $k$  objetos distintos  $\{i_1, i_2, \dots, i_k\}$  em que  $i_l$  é o maior objeto livre de peso no máximo igual a  $u_l$ , onde  $u_1 \leq u_2 \dots \leq u_k$  e  $w_{i_1} \leq \dots \leq w_{i_k}$ . Por exemplo, seja a instância definida por  $n = 7$  com pesos  $w = (0.9, 0.52, 0.47, 0.45, 0.3, 0.27, 0.21)$ . Então, o conjunto  $L[0.3, 0.4, 0.5]$  é formado pelos objetos  $i_1, i_2$  e  $i_3$  sendo  $i_1 = 6$  e  $w_6 = 0.27$  (maior peso menor ou igual a 3.0),  $i_2 = 5$  e  $w_5 = 0.3$  (maior peso menor ou igual a 4.0) e  $i_3 = 3$  e  $w_3 = 0.47$  (maior peso menor ou igual a 5.0).

Na primeira linha inicializa-se o número  $m$  de caixas usadas. A seguir, na linha 2, o conjunto  $N'$  guarda todos os objetos cujo peso é menor ou igual a  $1/5$ . Na linha 3, atualiza-se o conjunto de objetos livres  $N$  com aqueles cujo peso é maior do que  $1/5$ . As linhas 4-34 tratam os objetos com peso maior do que  $1/5$ . O primeiro laço (linhas 4-9) garante que as caixas geradas nesta etapa possuem peso menor ou igual a 1. No segundo laço (linhas 10-13), assume-se que existe um número par de objetos cujo peso pertence ao intervalo  $[0.5, 0.6)$ . Uma vez que qualquer objeto de cada par pesa menos do que  $3/5$ , as caixas pesarão no máximo  $6/5$ . Neste ponto do pseudo-código, só existem objetos com peso menor do que 0.5. O terceiro laço (linhas 14-17) cria caixas com peso, no máximo,  $0.3 + 0.4 + 0.5 = 6/5$ . Os objetos restantes pesam menos do que 0.5 e qualquer conjunto de dois objetos soma menos do que 1, nas caixas criadas no laço das linhas 18-21. As caixas criadas nas linhas 26-29 somam, no máximo,  $1 + 10 \cdot \delta/3$ . Uma vez que todos os objetos pesam mais do que  $1/5$ ,  $\delta < 1/4 - 1/5 = 1/20$  e tem-se que  $1 + 10 \cdot \delta/3 < 7/6 < 6/5$ . As caixas com três objetos, criadas nas linhas 30-33, não podem exceder  $3 \cdot 0.4 = 1.2$ . Finalmente, o último laço (linhas

35-40) trata os objetos cujo peso é menor ou igual a  $1/5$ . Empacotam-se todos os objetos restantes em qualquer caixa com peso inferior a 1.2 ou, se tal caixa não existir, em uma nova caixa. Para completar a descrição do algoritmo, deve-se considerar o caso em que o número de objetos com pesos  $w_i \in [0.5, 0.6)$ , na linha 10, é ímpar. Considera-se um único objeto restante  $i$  com  $w_i \in [0.5, 0.6)$ . Ele deve ser empacotado em uma caixa com no máximo três objetos, uma vez que todos os objetos pesam mais do que  $1/5$ . Deve-se então verificar e escolher qual das três opções a seguir cria a caixa mais pesada: empacotar  $i$  com  $L[0.25, 0.3]$ ; empacotar  $i$  com o objeto livre mais pesado; empacotar  $i$  sózinho em uma caixa não utilizada. A descrição acima mostra que nenhuma caixa pesará mais do que  $6/5$ . Como mencionado na Seção 4.2.1, Hochbaum e Shmoys [48] provam que o número de caixas utilizadas nunca ultrapassa o valor ótimo.

```

Procedimento 1/5_dual_BP
Entrada:     $N = \{1, \dots, n\}, w_i (i = 1, \dots, n)$ 
Saída:      $S = \{P_1, \dots, P_m\}, m.$ 
1   $m \leftarrow 0;$ 
2   $N' \leftarrow \{i \in N : w_i \leq 1/5\};$ 
3   $N \leftarrow N \setminus N';$ 
4  enquanto  $\exists i_a \in N$  tal que  $w_{i_a} \in [0.6, 1]$  faça
5       $m \leftarrow m + 1;$ 
6      se  $\exists i_1 \in L[1 - w_{i_a}] \subseteq N$  então
7           $P_m \leftarrow \{i_a, i_1\}; N \leftarrow N \setminus \{i_a, i_1\};$ 
8      senão  $P_m \leftarrow \{i_a\}; N \leftarrow N \setminus \{i_a\};$ 
9  fim-enquanto
10 enquanto  $\exists i_a, i_b \in N, i_a \neq i_b$  tal que  $w_{i_a}, w_{i_b} \in [0.5, 0.6)$  faça
11      $m \leftarrow m + 1;$ 
12      $P_m \leftarrow \{i_a, i_b\}; N \leftarrow N \setminus \{i_a, i_b\};$ 
13 fim-enquanto
14 enquanto  $\exists i_1, i_2, i_3 \in L[0.3, 0.4, 0.5] \subseteq N$  e  $w_{i_3} \geq 0.4$  faça
15      $m \leftarrow m + 1;$ 
16      $P_m \leftarrow \{i_1, i_2, i_3\}; N \leftarrow N \setminus \{i_1, i_2, i_3\};$ 
17 fim-enquanto
18 enquanto  $\exists i_a \in N$  tal que  $w_{i_a} \in [0.4, 0.5)$  faça
19      $i_b \leftarrow \arg \max_{i \in N \setminus \{i_a\}} \{w_i\};$ 
20      $P_m \leftarrow \{i_a, i_b\}; N \leftarrow N \setminus \{i_a, i_b\};$ 
21 fim-enquanto
22 enquanto  $N \neq \emptyset$  faça
23      $i_a \leftarrow \arg \min_{i \in N} \{w_i\};$ 
24     se  $w_{i_a} \leq 0.25$  então faça
25          $\delta \leftarrow 0.25 - w_{i_a};$ 
26         se  $\exists i_1, i_2, i_3, i_4 \in L[0.25 + \delta/3, 0.25 + \delta, 0.25 + 3\delta] \subseteq N \setminus \{i_a\}$  então faça
27              $m \leftarrow m + 1;$ 
28              $P_m \leftarrow \{i_a, i_1, i_2, i_3\}; N \leftarrow N \setminus \{i_a, i_1, i_2, i_3\};$ 
29         fim-se
30     senão enquanto  $N \neq \emptyset$  faça
31          $m \leftarrow m + 1;$ 
32         Escolha  $i_a, i_b, i_c \in N; P_m \leftarrow \{i_a, i_b, i_c\}; N \leftarrow N \setminus \{i_a, i_b, i_c\};$ 
33     fim-se
34 fim-enquanto
35 para-todo  $i \in N'$  faça
36      $j \leftarrow 1;$ 
37     enquanto  $\sum_{h \in P_j} w_h + w_i > 1.2$  e  $j \leq m$  faça  $j \leftarrow j + 1;$ 
38     se  $j > m$  então  $m \leftarrow m + 1;$ 
39      $P_j \leftarrow P_j \cup \{i\}; N' \leftarrow N' \setminus \{i\};$ 
40 fim-para-todo
41 retorne
fim 1/5_dual_BP
    
```

Figura B.1: Pseudo-código do algoritmo 1/5-dual-BP