CMSC858F: NP Hardness via 3 Partition (Chapter 7)

Gihan Jayatilaka gihan@umd.edu

April 2022

1 Comments on the presentation

1.1 Definition 7.2.1

This is a repetition. I found the same text in **Definition 1.8.2**. In a way, it is good to have this on every chapter to make them self contained. However, I think we should use the word **recall** when we are repeating content. Same concern goes for **Figure 7.1**.

1.2 7.3 Partition Problems and Scheduling

"we'll introduce PARTITION (This is sometimes also called PARTITION)" is a little confusing. The reader might be wondering if this is a typo or not.

1.3 Exercise 7.3.9

If I am not mistake, we never formally defined what "Cook-Levin theorem" is in the introductory chapter. We talked about their results in an informal way.

1.4 Three Dimensional Matching (3DM) INSTANCE (Page 185)

It might be in the best interests of the book to not get into political/touchy subjects like the number of sexes.

1.5 Theorem 7.4.1

- The proof line is not justified properly. The text goes to the margin. This might be an issue if the book goes into print.
- "2nd, 3rd" should be "$2^{nd}, 3^{rd}$"

1.6 Page 188

"Packing Squares into a Rectangle SQ-RECT PACKING" should be **bold** as per the convention in the book.

1.7 SIGNED EDGE MATCHING PUZZLE (SEMP)

- The question should end with a question mark (?).
- Here, we have used a, A in a particular font. On the next paragraph, we abuse the notation by writing it as a&A in a different font. I think it is good to keep these consistent.

1.8 General comment about punctuation

This chapter has multiple occurances of the following usage of full stops within brackets.

- Sentence (not a full sentence.)

This is wrong. The correct one should be either of the following.

- Full sentence. (full sentence.)
- Full sentence (not a full sentence).

1.9 Figure 7.21

The figure is hand drawn. Figure **??** can be used instead.

1.10 Figures 7.22-7.23

These figures are also hand drawn.

1.11 Theorem 7.7.5

Explaining what congruent is in brackets may be better.

2 Problems

All the problems listed in this section are proven to be NP hard by a reduction to 3-PARTITION problem.

2.1 Bin Packing Problem

[**?**] Consider n elements each with a positive size $S_1, S_2, ..., S_n$. Assume that there are unlimited bins of identical capacity C such that $C \geq max\{S_i\}$.

Problem: How to assign every S_i to bin such that the total number of bins used is minimum and no bin exceeds capacity?

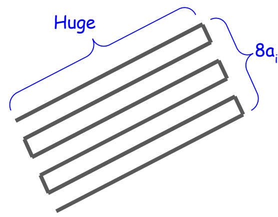

Figure 1: Figure 7.21 for the book

2.2 Minimum Common String Partition (MCSP) variant

[?]

Let $P = (P_i, P_2, \dots, P_m)$ be a partition of string X such that the concatenation of P_i's give X as $X = P_1, P_2, \dots, P_m$.

Similarly, $Q = (Q_1, Q_2, ...Q_m)$ is a partition of string Y.

$\pi = (P, Q)$ is a common partition of X, Y if Q is a permutation of P. This means, there exists a permutation σ such that $P_i = Q_{\sigma_i}$ for $i \in [m]$.

MCSP^C problem: Finding a π given two strings X, Y of length n over an alphabet of size c.

2.3 Scheduling Problems for Parallel/Pipelined Machines

[?]

Consider two processors P_1, P_2 which each can computer jobs of types $1, 2$ respectively.

Let there be n jobs $J = \{J_1, J_2, , , , J_n\}$ where each job is characterized by $J_i = (K_i, T_i, D_i, R_i)$ where K_i is the job type, T_i is the execution time in unitary representation, D_i delay time in pipeline processor architecture and R_i resource requirement.

Let $G = (J, E)$ be the precedence graph which specifies which job should be executed before the other one.

Let the schedule S be an ordering of J.

Problem: Find S to minimize the total completion time of all jobs.

2.4 A note on the complexity of the concurrent open shop problem

[?]

Consider m machines $(M_1, M_2, ..., M_m)$ and n jobs $(J_1, J_2, ..., J_n)$.

Each job consists of m different components at most and each component can be computed by a specific machine. Components are independent and they can run in parallel on different machines. A job is completed if all components have finished running.

Every job J_i comes to the system at a job release date R_i. There is a weight for how important a job is w_i.

Job J_i's completion time is C_i.

Problem: How to utilize the machines to minimize the weighted job completion time $\sum_i^n w_i C_i$?

2.5 Scheduling jobs with position-dependent processing times

[?]

Consider a single machine and a set of n jobs $J = (J_1, J_2, ..., J_n)$ and their ready times $R_1, R_2, ...R_n$.

Each job can have a processing time $p_i(v)$ in one of the form $a_i v^{-b}$ where a, b are constants.

Let π be a permutation of $[n]$ which is the ordering of jobs.

The completion time of a job is $C_{\pi(i)}$.

Problem: Find the optimal π to minimize the $max\{C_{\pi(i)}\}$.

2.6 Measurement Errors Make the Partial Digest Problem NP-Hard

[?]

Let m be an integer. Let D be a multi set of ${}^mC_2$ integers. Let δ be a positive integer. Let P be a set of m points.

Problem: Can P be on a line such that D is the distance multiset of the points up to an additive error of δ?

Note: There is a similar problem for multiplicative errors.

2.7 Fast balanced partitioning is hard even on grids and trees

[?]

Consider a graph $G = (V, E)$ with n vertices and an integer k.

Let G be a solid grid graph.

Problem: How to cut G into k sets with each set having at most $\lceil \frac{n}{k} \rceil$ nodes per set with an ϵ approximation factor?

Note: There is a similar problem where G is a tree.

2.8 The VLSI layout problem in various embedding models

[?]

Consider a graph G with degree bound 4 and a positive integer A.

Problem: Can we embedd G into a grid such that $area \leq A$?

2.9 Strong NP-completeness of a matrix similarity problem

[?] Consider a upper traingular matrix $A \in R^{n \times n}$. The diagonal elements of A are distinct.

Let $\tau \geq 1$ be a constant.

Let G be a non singular matrix with a condition number bounded by τ.

Problem: Is there a G such that $G^{-1}AG$ is a 2×2 block diagonal.

2.10 The train positioning problem

[?] This is a application oriented paper. It describes a situation where cars are being loaded to a set of trains using some cranes.

Consider a set of trains $T = (T_1, T_2 ... T_n)$. Trains are placed along the x axis with lengths $L_1, L_2 ..., L_n$ and start positions $x_{sT_1}, x_{sT_2} ..., x_{sT_n}$.

$L_i = \|x_{eT_1} - x_{sT_1}\|$

Let there be m cranes such that they cover a particular x range for loading cars.

Problem: How to place trains (pick x_{sT_i}'s) such that the maximum number of cars to be loaded by a crane is minimum?

References

[1] Mauro Dell'Amico and Silvano Martello. Reduction of the three-partition problem. *Journal of combinatorial optimization*, 3(1):17–30, 1999.

[2] Haitao Jiang, Binhai Zhu, Daming Zhu, and Hong Zhu. Minimum common string partition revisited. *Journal of Combinatorial Optimization*, 23(4):519–527, 2012.

[3] David Bernstein, Michael Rodeh, and Izidor Gertner. On the complexity of scheduling problems for parallel/pipelined machines. *IEEE Transactions on computers*, 38(9):1308–1313, 1989.

[4] Thomas A Roemer. A note on the complexity of the concurrent open shop problem. *Journal of scheduling*, 9(4):389–396, 2006.

[5] Aleksander Bachman and Adam Janiak. Scheduling jobs with position-dependent processing times. *Journal of the Operational Research Society*, 55(3):257–264, 2004.

[6] Mark Cieliebak and Stephan Eidenbenz. Measurement errors make the partial digest problem np-hard. In *Latin American Symposium on Theoretical Informatics*, pages 379–390. Springer, 2004.

[7] Andreas Emil Feldmann. Fast balanced partitioning is hard even on grids and trees. *Theoretical Computer Science*, 485:61–68, 2013.

[8] Michael Formann and Frank Wagner. The vlsi layout problem in various embedding models. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 130–139. Springer, 1990.

[9] V Brimkov, Bruno Codenotti, Mauro Leoncini, and Giovanni Resta. Strong np-completeness of a matrix similarity problem. *Theoretical computer science*, 165(2):483–490, 1996.

[10] Dirk Briskorn and Malte Fliedner. The train positioning problem. *Maxim Sviridenko, IBM Watson Research Center Steef van de Velde, Erasmus University Rotterdam Gerhard Woeginger, Technische Universiteit Eindhoven Guochuan Zhang, Zhejiang University*, page 231, 2011.