

# How to Use SFM

onigiri

November 3, 2015

## Contents

<b>1</b>	<b>Options</b>	<b>3</b>
1.1	Oracle . . . . .	3
1.2	Submodular Functions . . . . .	3
1.3	Input File . . . . .	4
1.4	Output File . . . . .	4
1.5	Numerical Error Tolerance . . . . .	4

# 1 Options

We need to set some options in order to execute SFM.exe.

## 1.1 Oracle

We can set a submodular function oracle by "-o" option.

- 0: Undirected Cut
- 1: Directed Cut
- 2: Connected Detachment
- 3: Facility Location
- 4: Graphic Matroid
- 5: Set Cover
- 6: Non Positive Symmetric Matrix Summation
- 7: Binary Matroid
- 8: Set Cover Concave

See "Document

DataCreator

DataCreator.pdf" for details. In order to use Set Cover Concave, we should also set parameter "-p" and "-c" as  $\alpha$  and  $c$  in the pdf file, respectively. Note that we can only use Undirected Cut, Directed Cut, Set Cover and Set Cover Concave when we use Fujishige-Wolfe type algorithm explained next section.

## 1.2 Submodular Functions

We can set an algorithm for submodular function minimization by "-a" option. ("-a 8" and "-a 100" are logically the same algorithms, but implemented by slightly different way.)

- 0: Iwata-Fleischer-Fujishige weakly polynomial time algorithm in [4].
- 1: Iwata-Fleischer-Fujishige Strongly polynomial time algorithm in [4].
- 2: Schrijver's algorithm in [7].
- 3: Iwata's hybrid weakly polynomial time algorithm in [3].
- 4: Iwata's hybrid strongly polynomial time algorithm in [3].
- 5: Orlin's algorithm in [6].
- 6: Iwata-Orlin weakly polynomial time algorithm in [5].

- 7: Iwata-Orlin strongly polynomial time algorithm in [5].
- 8: Fujishige-Wolfe algorithm without any reduction approach in [1].
- 100: Fujishige-Wolfe algorithm without any reduction approach in [1].
- 101: Fujishige-Wolfe algorithm with reduction in [2]
- 102: Fujishige-Wolfe algorithm improved by Kitamura Masashi.

Not that when we use weakly polynomial time algorithm, the program assume that submodular functions are integer-valued. Therefore, if we use a real-valued submodular function, the program may not obtain a correct answer. We can only use Undirected Cut, Directed Cut, Set Cover and Set Cover Concave when we use Fujishige-Wolfe type algorithm (8, 9, 10).

### 1.3 Input File

We can set the file for the input data by "-f" option. See "Document DataCreator DataCreator.pdf" for the format of input data.

### 1.4 Output File

We can set the file for writing output by "-r" option.

### 1.5 Numerical Error Tolerance

We can set the numerical error tolerance by "-e" option. We do not need to set this value and in that case, the tolerance is  $10^{-10}$ . Note that if we use severe error tolerance or non-integer submodular function, it is possible for the program to take huge time.

## References

- [1] S. Fujishige. Submodular systems and related topics. *Mathematical Programming Study*, 22:113–131, 1984.
- [2] S. Fujishige and S. Isotani. A submodular function minimization algorithm based on the minimum-norm base. *Pacific Journal of Optimization*, 7(1):3–17, 2011.
- [3] S. Iwata. A faster scaling algorithm for minimizing submodular functions. *SIAM Journal on Computing*, 32(4):833–840, 2003.
- [4] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4):761–777, 2001.

- [5] S. Iwata and J. B. Orlin. A simple combinatorial algorithm for submodular function minimization. In *Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms*, pages 1230–1237. Society for Industrial and Applied Mathematics, 2009.
- [6] J. B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009.
- [7] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.