Applications and Solutions of the Vertex Seperator Problem (VSP)

Ovidiu Nechita
onechita@ufl.edu

Vertex Seperator Problem (VSP)[1]

- We are given a connected graph, where each node i has a weight w_i and a cost c_i
- Partition a given graph into 3 parts $\mathcal{A}, \mathcal{B}, C$ so that no edges go from $\mathcal{A}$ to $\mathcal{B}$
- We seek to minimize the cost of the separator
- Don't have $\mathcal{A}$ or $\mathcal{B}$ too big. $\ell_{\mathcal{A}}$ and $u_{\mathcal{B}}$ are upper and lower bounds on the weight of $\mathcal{A}$. And $\ell_{\mathcal{B}}$ and $u_{\mathcal{B}}$ are upper and lower bounds on the weight of $\mathcal{B}$.

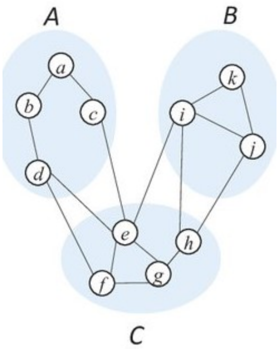


Figure: In the graph above, group C is a separator of $\mathcal{A}$ and $\mathcal{B}$

[1] Fuda Ma, Yang Wang, and Jin-Kao Hao. "Path relinking for the vertex separator problem". In: Expert Systems with Applications 82 (Mar. 2017). DOI: 10.1016/j.eswa.2017.03.064.

VSP Example

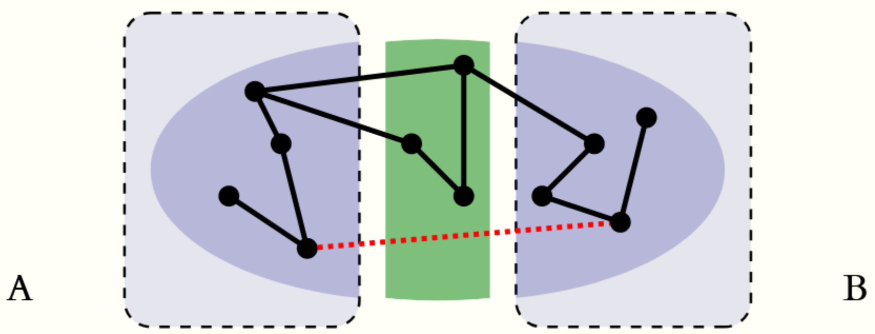


Figure: In this graph, the set of vertices in the middle is a vertex separator. Removing this set breaks the graph into two parts $\mathcal{A}$ and $\mathcal{B}$ that are not connected by an edge. If we added, the red edge to the graph, the set of vertices in the middle would no longer be a vertex separator.[2]

[2] Boaz Barak and David Steurer. Arora–Rao–Vazirani approximation for expansion. URL: https://www.sumofsquares.org/public/lec-arv.html.

Applications of the VSP

- Sparse Matrix Factorization[3]
- Parallel and distributed computing - the VSP can be used for hypergraph partitioning[4]
- Cyber security & Telecommunication networks - a separator determines network brittleness[5]
- Bioinformatics and computational biology[6]
- Many graph algorithms, especially those based on divide-and-conquer[7]

[3] XIE Xian-fen GU Wan-rong HE Yi-chen MAO Yi-jun. "Matrix Transformation and Factorization Based on Graph Partitioning by Vertex Separator for Recommendation". In: Computer Science 49.6A, 272 (2022), p. 272. DOI: 10.11896/jsjkx.210600159. URL: https://www.jsjkx.com/EN/abstract/article_20814.shtml.

[4] Enver Kayaaslan et al. "Partitioning Hypergraphs in Scientific Computing Applications through Vertex Separators on Graphs". In: SIAM Journal on Scientific Computing 34.2 (2012), A970–A992. DOI: 10.1137/100810022. eprint: https://doi.org/10.1137/100810022. URL: https://doi.org/10.1137/100810022.

[5] Charles E. Leiserson. "Area-efficient graph layouts". In: 21st Annual Symposium on Foundations of Computer Science (sfcs 1980). 1980, pp. 270–281. DOI: 10.1109/SFCS.1980.13.

[6] Bin Fu and Zhixiang Chen. "Sublinear Time Width-Bounded Separators and Their Application to the Protein Side-Chain Packing Problem". In: Algorithmic Aspects in Information and Management. Ed. by Siu-Wing Cheng and Chung Keung Poon. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 149–160.

[7] Cem Evrendilek. "Vertex separators for partitioning a graph". In: Sensors 8.2 (2008), pp. 635–657.

Multilevel Framework

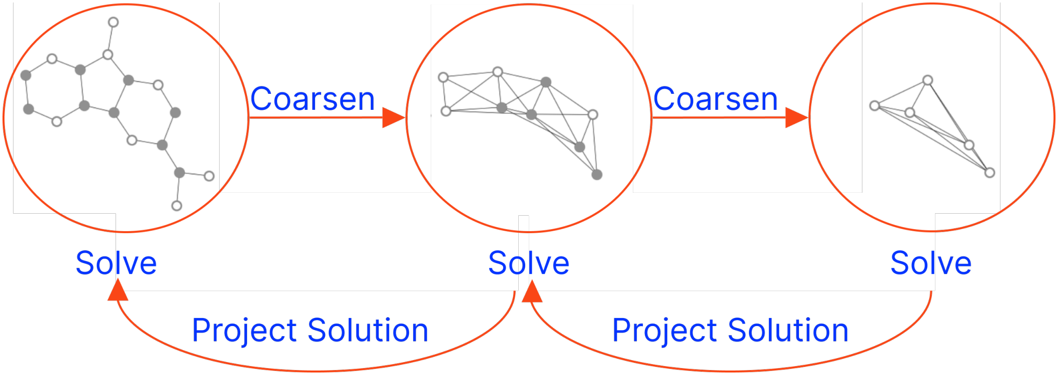


Figure: An illustration of the graph coarsening procedure[8]

[8] Tengfei Ma and Jie Chen. "Unsupervised Learning of Graph Hierarchical Abstractions with Differentiable Coarsening and Optimal Transport". In: CoRR abs/1912.11176 (2019). arXiv: 1912.11176. URL: http://arxiv.org/abs/1912.11176.

Solving One Level Of Coarsening

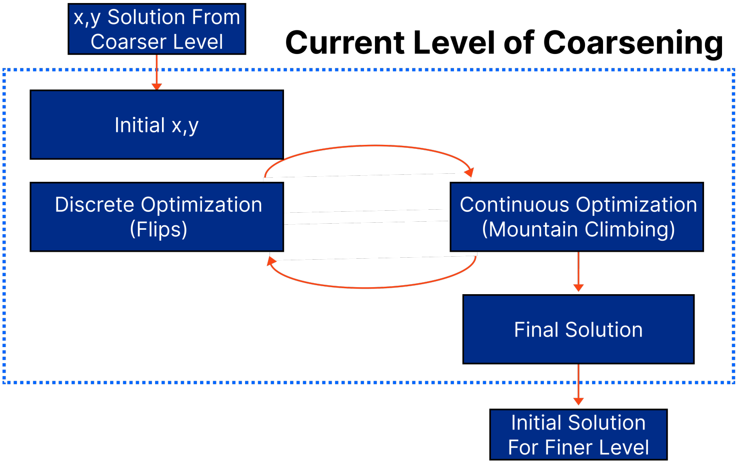


Figure: An illustration of how one level of coarsening is solved.

Initial Formulation

- For any $U \subseteq V$, define:
  - $\mathcal{C}(U) = \sum_{v \in U} c_v$
  - $\mathcal{W}(U) = \sum_{v \in U} w_v$
- Formulation:

$$\begin{aligned} \min_{\mathcal{A},\mathcal{B},S} \ & \mathcal{C}(S) \\ \text{s.t. } & \mathcal{A} \cap \mathcal{B} = \emptyset \\ & (\mathcal{A} \times \mathcal{B}) \cap E = \emptyset \\ & \ell_A \leq \mathcal{W}(\mathcal{A}) \leq u_A \\ & \ell_B \leq \mathcal{W}(\mathcal{B}) \leq u_B \\ & S = V - (\mathcal{A} \cup \mathcal{B}) \end{aligned} \iff \begin{aligned} \max_{\mathcal{A},\mathcal{B}} \ & \mathcal{C}(\mathcal{A} \cup \mathcal{B}) \\ \text{s.t. } & \mathcal{A} \cap \mathcal{B} = \emptyset \\ & (\mathcal{A} \times \mathcal{B}) \cap E = \emptyset \\ & \ell_A \leq \mathcal{W}(\mathcal{A}) \leq u_A \\ & \ell_B \leq \mathcal{W}(\mathcal{B}) \leq u_B \end{aligned}$$

Notation

- With the subset $\mathcal{A}$, we associate the binary vectors x, y:

$$x_i = \begin{cases} 1 & \text{if } i \in \mathcal{A} \\ 0 & \text{if } i \notin \mathcal{A} \end{cases} \qquad y_i = \begin{cases} 1 & \text{if } i \in \mathcal{B} \\ 0 & \text{if } i \notin \mathcal{B} \end{cases}$$

- Let A be the adjacency matrix.
- The condition $\mathcal{A} \cap \mathcal{B} = \emptyset$ is captured by the equation $x^T y = 0$
- The condition $(\mathcal{A} \times \mathcal{B}) \cap E = \emptyset$ is captured by $x^T A y = 0$.
- Adding them, we get $x^T (A + I) y = 0$.

Binary Vector Formulation

- Let $\mathbb{B} = \{0, 1\}$.

$$
\begin{aligned}
\max_{\mathcal{A},\mathcal{B}} \; & \mathcal{C}(\mathcal{A} \cup \mathcal{B}) \\
\text{s.t.} \; & \mathcal{A} \cap \mathcal{B} = \emptyset \\
& (\mathcal{A} \times \mathcal{B}) \cap E = \emptyset \\
& \ell_A \le \mathcal{W}(\mathcal{A}) \le u_A \\
& \ell_B \le \mathcal{W}(\mathcal{B}) \le u_B
\end{aligned}
\quad \iff \quad
\begin{aligned}
\max \; & c^T(x+y) \\
\text{s.t.} \; & x^T(A+I)y = 0 \\
& \ell_{\mathcal{A}} \le w^T x \le u_{\mathcal{A}} \\
& \ell_{\mathcal{B}} \le w^T y \le u_{\mathcal{B}} \\
& x, y \in \mathbb{B}^n
\end{aligned}
$$

- The constraint $x^T(A+I)y = 0$ is somewhat complex and difficult to satisfy.
- We bring it into the objective, with a penalty $\gamma > 0$.

Relaxing a Constraint

$$\begin{aligned} \max\ & c^T(x+y) \\ \text{s.t. } & x^T(A+I)y = 0 \\ & \ell_{\mathcal{A}} \le w^T x \le u_{\mathcal{A}} \\ & \ell_{\mathcal{B}} \le w^T y \le u_{\mathcal{B}} \\ & x, y \in \mathbb{B}^n \end{aligned} \iff \begin{aligned} \max\ & c^T(x+y) - \gamma x^T(A+I)y \\ \text{s.t. } & \ell_{\mathcal{A}} \le w^T x \le u_{\mathcal{A}} \\ & \ell_{\mathcal{B}} \le w^T y \le u_{\mathcal{B}} \\ & x, y \in \mathbb{B}^n \end{aligned}$$

- **Proposition:** If $w \ge 1$ and $\gamma \ge \max\{c_i : i \in V\}$, then for any feasible point (x, y) of the relaxed problem satisfying $f(x, y) \ge \gamma(\ell_{\mathcal{A}} + \ell_{\mathcal{B}})$, there is a feasible point $(\bar{x}, \bar{y})$ of the strict problem which satisfies
$$c^T(\bar{x} + \bar{y}) \ge c^T(x + y) - \gamma x^T(A + I).$$
- In practice, $\ell_{\mathcal{A}} = \ell_{\mathcal{B}} = 1$ is common, so $f(x, y) \ge \gamma(\ell_{\mathcal{A}} + \ell_{\mathcal{B}})$ is easy to satisfy.

Relaxing the Binary Requirements

$$
\begin{aligned}
\max\ & c^T(x+y) - \gamma x^T(A+I)y \\
\text{s.t. } & \ell_{\mathcal{A}} \leq w^T x \leq u_{\mathcal{A}} \\
& \ell_{\mathcal{B}} \leq w^T y \leq u_{\mathcal{B}} \\
& x, y \in \mathbb{B}^n
\end{aligned}
\iff
\begin{aligned}
\max\ & c^T(x+y) - \gamma x^T(A+I)y \\
\text{s.t. } & \ell_{\mathcal{A}} \leq w^T x \leq u_{\mathcal{A}} \\
& \ell_{\mathcal{B}} \leq w^T y \leq u_{\mathcal{B}} \\
& 0 \leq x, y \leq 1
\end{aligned}
$$

- **Proposition:** If the original VSP problem is feasible, then the continuous problem has a mostly binary solution. (i.e. at most one component in each of x, y is fractional)[9]

[9] William W. Hager, James T. Hungerford, and Ilya Safro. “A multilevel bilinear programming algorithm for the vertex separator problem”. In: Computational Optimization and Applications 69.1 (2018), pp. 189–223. ISSN: 1573-2894. DOI: 10.1007/s10589-017-9945-2. URL: https://doi.org/10.1007/s10589-017-9945-2.

Flips

$$
\begin{aligned}
\min\ & \gamma x^T(A+I)y - c^T(x+y) \\
\text{s.t.}\ & w^T x \le u_{\mathcal{A}} \\
& w^T y \le u_{\mathcal{B}} \\
& 0 \le x, y \le 1
\end{aligned}
$$

- **Single Flips:** Flip a single entry in x or y from 0 to 1 (up-flip) or 1 to 0 (down-flip).
- **Mixed Flips:** Simultaneous up-flip in a vector and down-flip in the other.

Possible Flips

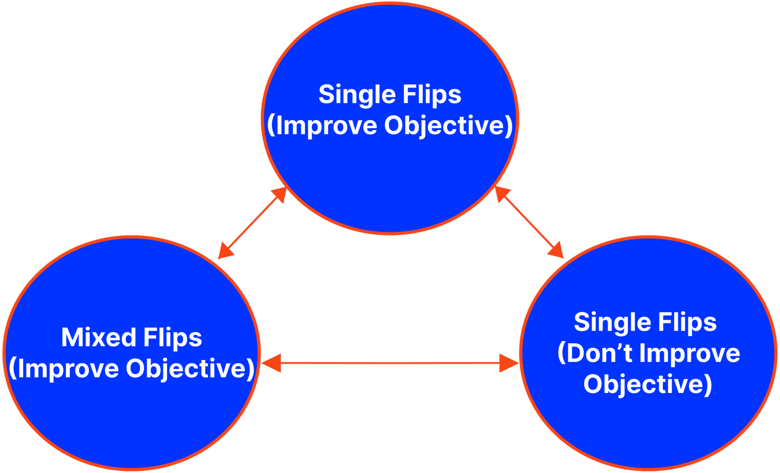
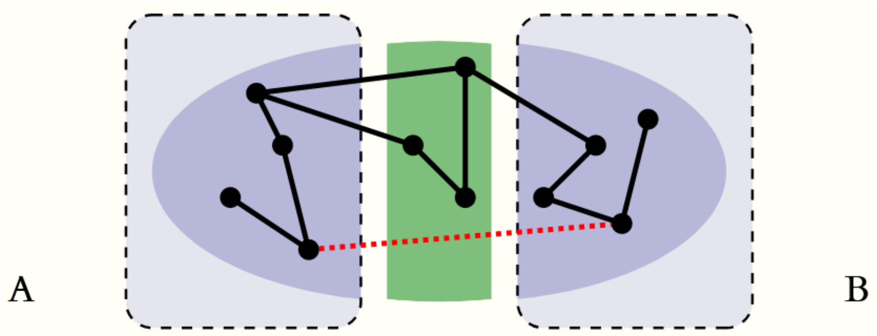


Figure: An illustration of the possible flips that can be selected.

Example Flips

- How do we find the best possible flip?



Variation of Konno's Mountain Climbing Algorithm[10]

$$\min_{x,y} f(x,y) \quad s.t. \quad x \in P_1, y \in P_2$$

Algorithm Modified Mountain Climbing Algorithm

Data: A feasible point (x, y)

while (No more improvement in objective possible) **do**

  $x^* \leftarrow \text{argmin}\{f(x,y) : x \in P_1\}$

  $y^* \leftarrow \text{argmin}\{f(x,y) : y \in P_2\}$

  Find the minimum of $\{f(x^*,y), f(x,y^*), f(x^*,y^*)\}$

  Update (x,y) to either $(x^*,y), (x,y^*)$, or (x^*,y^*) accordingly

end

[10] Hiroshi Konno. "A cutting plane algorithm for solving bilinear programs". In: Mathematical Programming 11.1 (1976), pp. 14–27. ISSN: 1436-4646. DOI: 10.1007/BF01580367. URL: https://doi.org/10.1007/BF01580367.

Mountain Climbing Applied to VSP

$$
\begin{aligned}
\min\ & \gamma x^T(A+I)y - c^T(x+y) \\
\text{s.t.}\ & w^T x \le u_{\mathcal{A}} \\
& w^T y \le u_{\mathcal{B}} \\
& 0 \le x, y \le 1
\end{aligned}
$$

- Constraints are decoupled, so we apply Mountain Climbing
- Objective is linear in x

$$
\begin{aligned}
\min\ & \gamma x^T(A+I)y - c^T(x+y) \\
\text{s.t.}\ & w^T x \le u_{\mathcal{A}} \\
& 0 \le x \le 1
\end{aligned}
\quad \Longleftrightarrow \quad
\begin{aligned}
\min\ & d^T x \\
\text{s.t.}\ & w^T x \le u_{\mathcal{A}} \\
& 0 \le x \le 1
\end{aligned}
$$

Solving the LP

$$\begin{aligned} \min\ & d^T x \\ \text{s.t.}\ & w^T x \leq u_{\mathcal{A}} \\ & 0 \leq x \leq 1 \end{aligned}$$

- Partial Lagrangian relaxation: for $0 \leq x \leq 1,\ \lambda \geq 0$,

$$L(x, \lambda) = d^T x + \lambda(w^T x - u_{\mathcal{A}}) = (d + \lambda w)^T x - \lambda u_{\mathcal{A}}$$

- Dual function: for $\lambda \geq 0$

$$L(\lambda) = \min_{0 \leq x \leq 1} (d + \lambda w)^T x - \lambda u_{\mathcal{A}}$$

Finding the Dual

- Dual function: for $\lambda \geq 0$, $\quad L(\lambda) = \min_{0 \leq x \leq 1} (d + \lambda w)^T x - \lambda u_{\mathcal{A}}$
- A minimizer x^* is given by

$$x_i^* = \begin{cases} 1 & \text{if } \; d_i + \lambda w_i < 0 \\ 0 & \text{if } \; d_i + \lambda w_i > 0 \\ \text{free} & \text{if } \; d_i + \lambda w_i = 0 \end{cases}$$

- Therefore

$$L(\lambda) = \sum_{i : d_i + \lambda w_i < 0} (d_i + \lambda w_i) - \lambda u_{\mathcal{A}}$$

Solving the Dual

$$L(\lambda) = \sum_{i:d_i+\lambda w_i<0} (d_i + \lambda w_i) - \lambda u_{\mathcal{A}}$$

- The dual is a continuous, piecewise linear, concave function
- The points of non-differentiability occur at $\lambda = \frac{-d_i}{w_i}$

Figure: A Graph of the Dual Function

Next Steps & Conclusion

- The Flips and Mountain Climbing procedures have a very efficient implementation, and initial results suggest this algorithm is very fast.
- However, the algorithm does not yet produce very good separators.
- An extensive computational comparative analysis is needed to compare the algorithm's performance against state of the art algorithms for solving the VSP.

Next Steps & Conclusion Continued

- One issue which may contribute to the low-quality separators is that a large γ in the objective

$$\min \gamma x^T (A + I)y - c^T (x + y)$$

  heavily penalizes steps which violate the constraint $x^T (A + I)y$. This penalty may be keeping the algorithm from exploring other regions of the solution space which may have more favorable separators.

- Thus, a possible direction for research is reducing γ at key moments in the algorithm in order to allow the exploration of yet-unknown regions of the solution space.

Questions

Thank You!