Mixed Integer Linear Programming Approach for a Distance-Constrained Elementary Path Problem

Sébastien Francois, Rumen Andonov, Hristo Djidjev, Metodi Traikov, Nicola Yanev

Mixed Integer Linear Programming Approach for a Distance-Constrained Elementary Path Problem

Sebastien François[1], Rumen Andonov[1], Hristo Djidjev[2],
Metodi Traikov[3], Nicola Yanev[3]

[1] Univ Rennes, Inria, CNRS, IRISA, Rennes, France, sefra35@gmail.com, randonov@irisa.fr
[2] Los Alamos National Laboratory, Los Alamos, NM 87545, USA djidjev@lanl.gov
[3] Center for Advanced Bioinformatics Research, South-West University Neofit Rilski, Bulgaria
metodi.gt@gmail.com, choby@math.bas.bg

Abstract

Given a directed graph $G = (V, E, l)$ with weights $l_e \geq 0$ associated with arcs $e \in E$ and a set of vertex pairs with distances between them (called *distance constraints*), the problem is to find an elementary path in G that satisfies a maximum number of distance constraints. We describe two MIP formulations for this problem and discuss their advantages.

Keywords : *Graphs, Optimization, Longest Path Problem, Mixed Integer Programming*

1 Introduction

We are interested in the following problem. Consider a directed graph $G = (V, E, l)$ where $l_e \geq 0$ is the weight associated with each arc $e \in E$. It is also given a set DC of vertex pairs (called *distance constraints*) such that with any couple $(u, v) \in DC$ a distance interval $[\underline{d}(u, v), \overline{d}(u, v)]$ is associated (i.e. $(\underline{d}(u, v) \leq \overline{d}(u, v)$ are lower (upper) bounds for the distance between u and v along the solution path). Let $\widetilde{\mathcal{P}}$ be a path in G. We say that $\widetilde{\mathcal{P}}$ satisfies a given distance constraint $dc(u, v) = [\underline{d}(u, v), \overline{d}(u, v)]$ if both u and v are on $\widetilde{\mathcal{P}}$ and the subpath of $\widetilde{\mathcal{P}}$ between u and v has length in $dc(u, v)$. Recall that the length of a path p is defined as the sum of the weights of the arcs in p. Then the problem we are considering is to find an elementary[1] path in G that satisfies a maximum number of distance constraints. We call it *Distance-Constrained Elementary Path (DCEP)* problem. The DCEP is $\mathcal{NP}$-hard due to a simple reduction from the Hamiltonian path problem. To see this, consider a set DC containing all ordered pairs of vertices and such that $(\underline{d}(u, v), \overline{d}(u, v)) = (-\infty, \infty)$ for any $(u, v) \in DC$.

As far as we know, the DCEP problem has not been previously discussed in the combinatorial optimization community. The problem is motivated by applications in genome assembly in bioinformatics, and a variation of the problem has been originally described[2] in two of our previous publications [3, 4]. The *Genome assembly* problem is a challenging computational task aiming at reconstructing the full genome of an organism from short DNA sequences (*reads*) [6]. Note that in spite of the efforts and the progress done by the bioinformatics community, no satisfactory solution is available today. The method proposed in [3, 4] is based on integer programming model for solving genome assembly as a problem of finding a long simple path in a specific graph, which satisfies additional constraints encoding the insert-size (distance) information. This methodology significantly differs from the heuristics described in the literature like BESST [8] and SPAdes [1]. The computations performed on

[1] An elementary (also called simple) is a path that visits each vertex at most once.
[2] However, the name Distance-Constrained Elementary Path is firstly used here.

several chloroplast genomes demonstrate that such an approach outperforms these widely-used assembly solvers by the accuracy of the results.

While the paper [4] is application oriented, here we revisit the mixed integer linear programming formulation proposed there from a combinatorial optimization viewpoint. Furthermore, we show how to adapt the well known Miller, Tucker and Zemlin (MTZ) formulation for solving the longest path problem [7] in order to solve the DCEP problem. Note that the challenges in DCEP problem are somehow similar, but harder in practice, to the ones in solving longest/shortest (with real weights) elementary path problems [2, 9]).

2 Standard integer programming formulation related to elementary path description in a directed graph

One of the specificities of the DCEP problem is that the beginning and the end of the path are unknown. The most natural approach to overcome this problem is to introduce two dummy vertices s and t (source and target) and to connect them with the rest of the vertices. The extended graph has $2|V|$ more edges. Hence we consider a directed graph $\hat{G} = \{\hat{V}, \hat{E}, l\}$ with set of nodes $\hat{V}$ and set of arcs $\hat{E}$ where

- $\hat{V} = V \cup \{s, t\}$ and s and t are such that $s \notin V$ and $t \notin V$.
- $\hat{E} = E \cup \{(s, v)|v \in V\} \cup \{(u, t)|u \in V\}$ and $l(s, v) = l(u, t) = 0, (s, v) \in \hat{E}, (u, t) \in \hat{E}$.

In all formulations below we assume that $|\delta^-(s)| = |\delta^+(t)| = \emptyset$, where by $\delta^-(v)/\delta^+(v)$ we denote the set of ingoing/outgoing edges for a vertex v. The standard integer programming formulation for constructing an elementary path from s to t is to find values of binary variables x_e for all $e \in \hat{E}$ (whose meaning is that $x_e = 1$ if e belongs to the solution path, and $x_e = 0$, otherwise) such that

$$\forall u \in V, \sum_{e \in \delta^+(u)} x_e \leq 1 \tag{1}$$

$$\forall u \in V' : \sum_{e \in \delta^+(u)} x_e - \sum_{e \in \delta^-(u)} x_e = \begin{cases} 1 & \text{if u} = s \\ -1 & \text{if u} = t \\ 0 & \text{else} \end{cases} \tag{2}$$

Constraint (1) ensure that the outgoing degree of each node is at most one, and constraints (2) are flow conservation constraints.

In the sequel we describe two different sets of constraints and variables that can be added to formulation (1)-(2) for sub-tour elimination in case of cycles.

2.1 Sequential formulation (MTZ) and its adaptation to the DCEP

To derive the well known MTZ formulation ([7]), it is enough to introduce, for each vertex v, an auxiliary variable $y_v \geq 0$ that allows us to number/label the vertices along the path in an increasing order. The constraint (3) allows then to prevent sub-tours.

$$\forall (u, v) \in E, (y_v - y_u) \geq x_{u,v} - (1 - x_{u,v})|V|. \tag{3}$$

Set $W = \sum_{e \in E} l_e$ and $y_s = W$. To adapt (3) to the DCEP we replace (3) by the constraint

$$\forall (u, v) \in \hat{E},\ y_v \leq y_u - x_{u,v} l_{u,v} + (1 - x_{u,v})W, \tag{4}$$

where W plays the role of a big constant. Since $l_e \geq 0$, the labels y_v are decreasing along the path. This avoids cycles. Another advantage of (4) is that, for any couple (u, v) from the path, the gap $y_u - y_v$ measures the distance between vertices u and v.

In order to manage the distance constraints, we apply a technique similar to the one in [3, 4]. We introduce a new variable $g_e \in \{0, 1\}, e \in DC$, and we set to 1 the value of $g_{(u,v)}$ if and

only if both vertices u and v belong to the selected path and its length between them is in the given interval $[\underline{d}_{(u,v)}, \overline{d}_{(u,v)}]$. We also add the constraint

$$\forall v \in V,\ y_v \leq W(\sum_{e \in \delta^-(v)} x_e), \tag{5}$$

which sets to zero the values of y_v outside the selected path. The other constraints related to the distances DC are

$$\forall e \in DC : g_{(u,v)} \leq y_u \text{ and } g_{(u,v)} \leq y_v \tag{6}$$

as well as

$$\forall (u,v) \in DC : \quad \overline{d}(u,v) g_{(u,v)}) + W(1 - g_{(u,v)}) \ \geq y_u - y_v \geq \ \underline{d}(u,v) g_{(u,v)} - W(1 - g_{(u,v)}). \tag{7}$$

We search for a path that satisfies a maximum number of distance constraints

$$\max W(\sum_{e \in DC} g_e) + \sum_{v \in V} y_v. \tag{8}$$

The last term in (8) forces the labels of the vertices on the path to take their minimal values. We denote by MTZDC the obtained in this manner formulation.

3 GAT formulation ([4])

To any vertex $v \in V$ we associate a variable $i_v,\ 0 \leq i_v \leq 1$, encoding whether v is on the solution path. The two possibles states for a vertex v (to be an intermediate vertex in the path or not) are enforced by the following constraints

$$i_v = \sum_{e \in \delta^+(v)} x_e = \sum_{e \in \delta^-(v)} x_e. \tag{9}$$

One can then show [5] that the real variables $i_v, \forall v \in V$ take binary values.

We introduce a continuous variable $f_e \in R^+$ to express the quantity of the flow circulating along the edge $e \in E$. Without this variable, the solution found may contain some loops and hence may not be a simple path. We put a requirement that no flow can use an edge e when $x_e = 0$, which can be encoded as

$$\forall e \in E : 0 \leq f_e \leq W x_e, \tag{10}$$

where W is as defined above ($W = \sum_{e \in E} l_e$). We use the flows f_e in the following constraint

$$\forall v \in V \ : \sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = \sum_{e \in \delta^+(v)} l_e x_e, \tag{11}$$

while for the source vertex we require $\sum_{e \in \delta^+(s)} f_e = W$. The flow then decreases along the path and this feature forbids cycles.

Furthermore, a variable $g_e \in \{0,1\}$ is associated with any distance constraint. The value of $g_{(u,v)}$ is set to 1 only if both vertices u and v belong to the selected path and its length between them is in the given interval $[\underline{d}_{(u,v)}, \overline{d}_{(u,v)}]$. The constraints are as follows :

$$\forall e \in DC : g_e \in \{0,1\} \text{ and } g_{(u,v)} \leq i_u \text{ and } g_{(u,v)} \leq i_v \tag{12}$$

as well as :

$$\forall (u,v) \in DC : \overline{d}(u,v) g_{(u,v)}) + M(1 - g_{(u,v)}) \geq \sum_{e \in \delta^-(u)} f_e - \sum_{e \in \delta^-(v)} f_e \ \geq \underline{d}(u,v) g_{(u,v)} - M(1 - g_{(u,v)}) \tag{13}$$

We search for a path that satisfies a maximum number of distance constraints.

$$\max \sum_{e \in DC} g_e. \tag{14}$$

4 Conclusion and perspectives

In Table 1 we summarize the presented formulations. We observe that, on theory, they are very similar–both formulations have almost the same number of variables and the same number of constraints. However, they are dual-like in the sense that in MTZDC the distances are computed in the vertex variables y_v, while in GAT they are in the flow variables f_e which are associated in the arcs. The sub-tour elimination constraint in MTZDC (4) requires $|E|$ inequalities, while the same constraints in GAT (11) requires $|V|$ equations. We have implemented both formulations with AMPL language. On small instances they behave similarly. We are currently getting statistics on huge instances towards more precise performance analysis.

	MTZDC			GAT		
	Name	Type	Number	Name	Type	Number
Variables	x_e	binary	|E|	x_e	binary	|E|
	y_v	real ≥ 0	|V|	i_v	$\geq 0, \leq 1$	|V|
				f_e	real ≥ 0	|E|
	g_e	binary	|DC|	g_e	binary	|DC|
	Purpose		Number	Purpose		Number
Constraints	Path		$2 \times \|V\|$	Path		$2 \times \|V\|$
	Subtour elimination		|E|	Subtour elimination		|V|
	Distances		$3 \times \|DC\|$	Distances		$3 \times \|DC\|$
	Variables bounding		|V|	Variables bounding		|E|

TAB. 1: A summary of the considered formulations

References

[1] A. Bankevich, S. Nurk, D. Antipov, A. Gurevich, M. Dvorkin, A. S. Kulikov, V. M. Lesin, S. I. Nikolenko, S. Pham, A. D. Prjibelski, A. V. Pyshkin, A. V. Sirotkin, N. Vyahhi, G. Tesler, M. A. Alekseyev, and P. A. Pevzner. Spades: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of computational biology : a journal of computational molecular cell biology*, 19(5):455–477, May 2012.

[2] Q. T. Bui, Y.c Deville, and Q. D. Pham. Exact methods for solving the elementary shortest and longest path problems. *Annals of Operations Research*, 244(2):313–348, 2016.

[3] S. François, R. Andonov, H. Djidjev, and D. Lavenier. Global optimization methods for genome scaffolding. *Electronic Notes in Discrete Mathematics*, 64, 2018.

[4] S. François, R. Andonov, D. Lavenier, and H. Djidjev. Global optimization approach for circular and chloroplast genome assemblies. In *10th International Conference on Bioinformatics and Computational Biology (BICoB 2018)*, 2018. Mars, 2018, Las Vegas, USA.

[5] Sebastien François, Rumen Andonov, Dominique Lavenier, and Hristo Djidjev. Global optimization methods for genome scaffolding. Technical Report 9050, INRIA, March 2017.

[6] Daniel H. Huson, Knut Reinert, and Eugene W. Myers. The greedy path-merging algorithm for contig scaffolding. *J. ACM*, 49(5):603–615, 2002.

[7] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *J. ACM*, 7(4):326–329, October 1960.

[8] K. Sahlin, F. Vezzi, B. Nystedt, J. Lundeberg, and L. Arvestad. BESST - efficient scaffolding of large fragmented assemblies. *BMC Bioinformatics*, 15:281, 2014.

[9] Leonardo Taccari. Integer programming formulations for the elementary shortest path problem. *European Journal of Operational Research*, 252(1):122–130, 2016.