Discrete Applied Mathematics ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

About the minimum mean cycle-canceling algorithm

Jean Bertrand Gauthier [a], Jacques Desrosiers [a,*], Marco E. Lübbecke [b]

[a] GERAD & HEC Montréal, 3000, chemin de la Côte-Sainte-Catherine, Montréal (Québec), Canada, H3T 2A7

[b] RWTH Aachen University, Kackertstraße 7, D-52072 Aachen, Germany

ARTICLE INFO

Article history:
Received 27 September 2013
Received in revised form 4 July 2014
Accepted 20 July 2014
Available online xxxx

Keywords:
Network flow problem
Residual network
Flow decomposition
Minimum mean cycle
Complexity analysis
Strongly polynomial algorithm

ABSTRACT

This paper focuses on the resolution of the capacitated minimum cost flow problem on a network comprising n nodes and m arcs. We present a method that counts imperviousness to degeneracy among its strengths, namely the *minimum mean cycle-canceling* algorithm (MMCC). At each iteration, primal feasibility is maintained and the objective function strictly improves. The goal is to write a uniform and hopefully more accessible paper which centralizes the ideas presented in the seminal work of Goldberg and Tarjan (1989) as well as the additional paper of Radzik and Goldberg (1994) where the complexity analysis is refined. Important properties are proven using linear programming rather than constructive arguments.

We also retrieve Cancel-and-Tighten from the former paper, where each so-called phase which can be seen as a group of iterations requires $O(m \log n)$ time. MMCC turns out to be a strongly polynomial algorithm which runs in $O(mn)$ phases, hence in $O(m^2 n \log n)$ time. This new complexity result is obtained with a combined analysis of the results in both papers along with original contributions which allows us to enlist Cancel-and-Tighten as an acceleration strategy.

1. Introduction

This paper addresses the resolution of the capacitated minimum cost flow problem (CMCF) on a network defined by n nodes and m arcs. We present the *minimum mean cycle-canceling* algorithm (MMCC). The seminal work of Goldberg and Tarjan [13], as presented in the book of Ahuja et al. [1], as well as the paper of Radzik and Goldberg [17], where the complexity analysis is refined, are the underlying foundations of this document. The current literature states that MMCC is a strongly polynomial algorithm that performs $O(m^2 n)$ iterations, a tight bound, and runs in $O(m^3 n^2)$ time.

While Goldberg and Tarjan [13] present Cancel-and-Tighten as a self-standing algorithm, we feel it belongs to the realm of acceleration strategies incidentally granting the reduction of the theoretical complexity. Our understanding is that this strategy can be shared at any level of the complexity analysis. Indeed, its very construction aims to assimilate the so-called notion of *phase* which can be seen as a group of iterations. This strategy exploits an approximation scheme to manage this assimilation and as such nevertheless necessitates a careful analysis. We propose a new approximation structure which allows us to reduce the global runtime to $O(m^2 n \log n)$. It is namely the product of a refined analysis that accounts for $O(mn)$ phases, each one requiring $O(m \log n)$ time.

The reader should view this work as much more than a synthesis. It is the accumulation of years of research surrounding degeneracy that led us to realize the ties with theories drafted some forty years ago. We not only hope to clarify the behavior

* Corresponding author.
E-mail addresses: jean-bertrand.gauthier@hec.ca (J.B. Gauthier), jacques.desrosiers@hec.ca (J. Desrosiers), marco.luebbecke@rwth-aachen.de (M.E. Lübbecke).

http://dx.doi.org/10.1016/j.dam.2014.07.005
0166-218X/© 2014 Published by Elsevier B.V.

of the minimum mean cycle-canceling algorithm but also provide strong insights about the ins and outs of its idiosyncrasies and more importantly establish a solid unified framework against which we can rest current and future work. On that note, let us underline the linear programming mindset which simplifies the construction of one of the most important parts of the algorithm, namely the pricing problem. The justification of some of its properties also benefit from straightforward implications provided by that mindset. Some fundamental properties of network problems are also incorporated throughout the text which sometimes facilitate if not, certainly enlighten, the comprehension of the proofs presented by the listed authors.

The paper is organized as follows. The elaboration of MMCC takes place in Section 2 where the combination of the so-called *residual network* along with optimality conditions give birth to a pricing problem which is put to use in an iterative process. Section 3 analyzes its complexity which is decomposed in two parts: the *outer loop* and the *bottleneck*. Although the latter comes at the very last, it acts as the binding substance of the whole paper. It is indeed where the behavior of the algorithm can be seen at a glance alongside the justification for the significance of the aforementioned *phases*. This is followed by the conclusion in Section 4.

2. Minimum mean cycle-canceling algorithm

Consider the formulation of CMCF on a directed graph $G = (N, A)$, where N is the set of n nodes associated with an assumed balanced set $b_i,\ i \in N$, of supply or demand defined respectively by a positive or negative value such that $\sum_{i\in N} b_i = 0$, A is the set of m arcs of cost $\mathbf{c} := [c_{ij}]_{(i,j)\in A}$, and $\mathbf{x} := [x_{ij}]_{(i,j)\in A}$ is the vector of bounded flow variables:

$$\begin{aligned} z^* := \quad & \min \sum_{(i,j)\in A} c_{ij}x_{ij} \\ \text{s.t.} & \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b_i, \qquad [\pi_i] \qquad && \forall i \in N \\ & 0 \le \ \ell_{ij} \le x_{ij} \le u_{ij}, && \forall (i,j) \in A, \end{aligned} \tag{1}$$

where $\boldsymbol{\pi} := [\pi_i]_{i\in N}$ is the vector of dual variables, also known as node potentials. When right-hand side $\mathbf{b} := [b_i]_{i\in N}$ is the null vector, formulation (1) is called a *circulation* problem.

Let us enter the world of network solutions with a fundamental proposition whose omitted proof traditionally relies on a constructive argument. It is so rooted in the network design that, case in point, straightforward derivatives are used throughout this document.

Proposition 1 (*Ahuja et al. [1, Theorem 3.5 and Property 3.6]*)**.** *Any feasible solution* $\mathbf{x}$ *to* (1) *can be represented as a combination of paths and cycles flows (though not necessarily uniquely) with the following properties:*

(a) *Every directed path with positive flow connects a supply node to a demand node; at most $n + m$ directed paths and cycles have non-zero flow among which at most m cycles.*
(b) *In the case of a circulation problem, by definition there are no supply nor demand nodes, which means the representation can be restricted to at most m directed cycles.*

This section derives MMCC, devised to solve instances of CMCF, in the following manner. Section 2.1 defines the corner stone of the resolution process, namely the residual network. Whether its inception goes back to the optimality conditions or its usage came as an afterthought is an enigma for which we have no answer. Either way, the latter are introduced thereafter and pave the way for the pricing problem in Section 2.2. Section 2.3 exhibits the algorithmic process which is ultimately information sharing between a control loop and a pricing problem. The former ensures primal feasibility while the latter provides a strictly improving direction at each iteration. Section 2.4 illustrates the behavior of the algorithm on the maximum flow problem.

2.1. Residual network and optimality conditions

The residual network takes form with respect to a feasible flow $\mathbf{x}^0 := [x^0_{ij}]_{(i,j)\in A}$ and is denoted $G(\mathbf{x}^0) = (N, A(\mathbf{x}^0))$. As eloquently resumed in Fig. 1, each arc $(i, j) \in A$ is replaced by two arcs representing upwards and downwards possible flow variations:

- arc (i, j) with cost $d_{ij} = c_{ij}$ and residual flow $0 \le y_{ij} \le r^0_{ij} := u_{ij} - x^0_{ij}$;
- arc (j, i) with cost $d_{ji} = -c_{ij}$ and residual flow $0 \le y_{ji} \le r^0_{ji} := x^0_{ij} - \ell_{ij}$.

Denote $A' := \{(i, j) \cup (j, i) \mid (i, j) \in A\}$ as the complete possible arc support of any residual network. The residual network $G(\mathbf{x}^0)$ consists of only the residual arcs, i.e., those with strictly positive residual capacities, that is, $A(\mathbf{x}^0) := \{(i, j) \in A' \mid r^0_{ij} > 0\}$. The combination of the current solution $\mathbf{x}^0$ along with the optimal marginal flow computed on the residual network is optimal for the original formulation. Indeed, the residual network with respect to $\mathbf{x}^0$ corresponds to the change of variables $x_{ij} = x^0_{ij} + (y_{ij} - y_{ji}),\ \forall (i, j) \in A$. Observe that traveling in both directions would be counterproductive and can be simplified

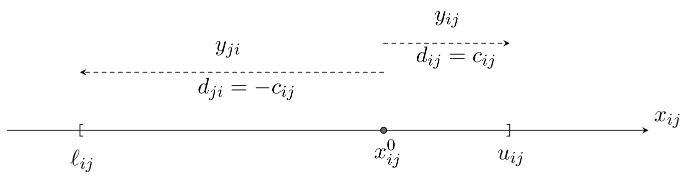

Fig. 1. A change of variables.

to the net flow in a single direction. This means that the marginal flow must be such that $y_{ij}\, y_{ji} = 0,\ \forall(i,j) \in A$, which is naturally verified by any practical solution.

Letting $z^0 = \mathbf{c}^\intercal \mathbf{x}^0$ means that CMCF can be reformulated as:

$$z^* := z^0 + \min \sum_{(i,j)\in A(\mathbf{x}^0)} d_{ij} y_{ij} \tag{2}$$

$$\text{s.t.} \sum_{j:(i,j)\in A(\mathbf{x}^0)} y_{ij} - \sum_{j:(j,i)\in A(\mathbf{x}^0)} y_{ji} = 0, \qquad [\pi_i] \qquad \forall i \in N \tag{3}$$

$$0 \le y_{ij} \le r^0_{ij}, \qquad \forall (i,j) \in A(\mathbf{x}^0). \tag{4}$$

Take a moment to consider an optimal solution of (2)–(4) on the residual network $G(\mathbf{x}^0)$. Mathematically speaking, it corresponds to a circulation problem. The right-hand side in (3) is zero everywhere, we are thus looking for a solution that respects the equilibrium already present in the current solution $\mathbf{x}^0$. Verifying that a directed cycle in $G(\mathbf{x}^0)$ exists as a cycle in G is as straightforward as applying the flow conservation principle, that is, we use the forward direction of arc (i,j) when $y_{ij} > 0$ or the backward direction when $y_{ji} > 0$.

Suppose that $\mathbf{x}$ and $\mathbf{x}^0$ are any two feasible solutions to formulation (1). Therefore some feasible circulation $\mathbf{y}$ in $G(\mathbf{x}^0)$ satisfies the property that $\mathbf{x} = \mathbf{x}^0 + \mathbf{y}$, and the cost of solution $\mathbf{x}$ is given by $\mathbf{c}^\intercal\mathbf{x} = \mathbf{c}^\intercal\mathbf{x}^0 + \mathbf{d}^\intercal\mathbf{y}$, where $\mathbf{d} := [d_{ij}]_{(i,j)\in A(\mathbf{x}^0)}$. Moreover, Proposition 1(b) means that there exists a way to decompose $\mathbf{y}$ in at most m cycles. We can therefore think of an optimal solution of (2)–(4) on $G(\mathbf{x}^0)$ as a collection of intertwined cycles. This is stated in the following proposition.

Proposition 2 (*Ahuja et al. [1, Theorem 3.7]*)**.** *Let* $\mathbf{x}$ *and* $\mathbf{x}^0$ *be any two feasible solutions of a network flow problem. Then* $\mathbf{x}$ *equals* $\mathbf{x}^0$ *plus the flow on at most m directed cycles in* $G(\mathbf{x}^0)$*. Furthermore, the cost of* $\mathbf{x}$ *equals* $\mathbf{c}^\intercal\mathbf{x}^0$ *plus the cost of flow on these augmenting cycles.*

Hence, there exists a way to move between any two feasible solutions in at most m cycles! It is quite a testament to how trivial reaching $\mathbf{x}^*$, *granted it is actually known*. The fact of the matter is that the residual network problem (2)–(4) is not easier to solve than the original problem (1). Nevertheless, there exists at least one sequence of transitions which constructs a series of residual networks allowing to move from $\mathbf{x}^0$ to an optimal solution $\mathbf{x}^*$ in a finite number of iterations.

It is indeed possible to think of the cycles contained in the residual network as transitioning possibilities. Consider the marginal changes instilled in $\mathbf{x}^0$ with respect to *some* negative (or improving) cycle and repeat this step until no such cycle remains. As simple as it may sound, we have stated the *generic cycle-canceling* algorithm as proposed by Klein [16], which ultimately amounts to a line search optimization method. Showing finiteness, at least as far as today's computer tractability is concerned (Ford and Fulkerson [10] show that pathological instances with irrational data could misbehave indefinitely or even worse converge to a bad solution), is as trivial as realizing this procedure performs a strict improvement in the objective function at each iteration until optimality is reached. However, it turns out that the order in which these cycles are identified has tremendous impact on the performance of this generic algorithm. Given integer data, denote the greatest absolute cost value by $C := \max_{(i,j)\in A} |c_{ij}|$ and the greatest interval range value by $U := \max_{(i,j)\in A} u_{ij} - \ell_{ij}$. Then the number of iterations of the generic algorithm ranges anywhere from $O(m)$ to $O(mCU)$.

Optimality conditions. With respect to $\boldsymbol{\pi}$, the reduced cost of variable x_{ij}, $(i,j) \in A$, is given by $\bar{c}_{ij} := c_{ij} - \pi_i + \pi_j$. Let the reduced cost $\bar{d}_{ij}$ of variable y_{ij}, $(i,j) \in A(\mathbf{x}^0)$, be computed in the same way, i.e., $\bar{d}_{ij} := d_{ij} - \pi_i + \pi_j$. For a feasible flow $\mathbf{x}^0$, we distinguish three equivalent necessary and sufficient optimality conditions. With respect to linear programming vocabulary, the first two can be qualified of primal and dual nature on the residual network $G(\mathbf{x}^0)$ while the third is that of complementary slackness on network G, see [1, Theorems 9.1, 9.3, and 9.4]:

Primal: $G(\mathbf{x}^0)$ contains no negative cycle.
Dual: $\exists\boldsymbol{\pi}$ such that $\bar{d}_{ij} \ge 0,\ \forall(i,j) \in A(\mathbf{x}^0)$.
Complementary slackness: $\exists\boldsymbol{\pi}$ such that, for every arc $(i,j) \in A$,

$$x^0_{ij} = \ell_{ij} \quad \text{if } \bar{c}_{ij} > 0; \qquad x^0_{ij} = u_{ij} \quad \text{if } \bar{c}_{ij} < 0; \qquad \bar{c}_{ij} = 0 \quad \text{if } \ell_{ij} < x^0_{ij} < u_{ij}. \tag{5}$$

Please cite this article in press as: J.B. Gauthier, et al., About the minimum mean cycle-canceling algorithm, Discrete Applied Mathematics (2014), http://dx.doi.org/10.1016/j.dam.2014.07.005

We underscore that if feasible flow $\mathbf{x}^0$ is actually optimal, all these conditions are verified simultaneously. Observe that the primal and dual conditions are only verifiable on the residual network $G(\mathbf{x}^0)$. The complementary slackness conditions however are verified on G by the combination of the current primal solution and the information gathered by the dual vector.

2.2. Pricing step: finding the minimum reduced cost

The pricing step elaborated in this section is derived from the residual network by capturing the rationale of the optimality conditions. According to the dual optimality condition, $\mathbf{x}^0$ is optimal if and only if there exists a vector $\boldsymbol{\pi}$ such that $d_{ij} - \pi_i + \pi_j \geq 0, \forall (i,j) \in A(\mathbf{x}^0)$. This can be verified by finding the smallest reduced cost, denoted μ^0, and formulated as the following linear program:

$$\mu^0 := \max \mu \tag{6}$$

$$\text{s.t. } \mu + \pi_i - \pi_j \leq d_{ij}, \qquad [y_{ij}] \quad \forall (i,j) \in A(\mathbf{x}^0). \tag{7}$$

Observe that $\boldsymbol{\pi}$ is not fixed but optimized in formulation (6)–(7). Its dual is expressed in terms of flow variables $[y_{ij}]_{(i,j)\in A(\mathbf{x}^0)}$:

$$\mu^0 := \min \sum_{(i,j)\in A(\mathbf{x}^0)} d_{ij} y_{ij} \tag{8}$$

$$\text{s.t. } \sum_{j:(i,j)\in A(\mathbf{x}^0)} y_{ij} - \sum_{j:(j,i)\in A(\mathbf{x}^0)} y_{ji} = 0, \quad [\pi_i] \qquad \forall i \in N \tag{9}$$

$$\sum_{(i,j)\in A(\mathbf{x}^0)} y_{ij} = 1, \quad [\mu] \tag{10}$$

$$y_{ij} \geq 0, \qquad \forall (i,j) \in A(\mathbf{x}^0), \tag{11}$$

where $\boldsymbol{\pi}$ is associated with the flow conservation constraints (9) while μ is a dual scalar associated with the convexity constraint (10). We already know optimality conditions provide alternative ways to prove the optimality status of feasible solution $\mathbf{x}^0$. It might not be all that surprising that if pricing problem (6)–(7) expresses the dual optimality condition on $G(\mathbf{x}^0)$, formulation (8)–(11) echoes the verification of the primal optimality condition on the residual network. Indeed, the latter is known as the *minimum mean cycle* problem, arguably giving all meaning to the algorithm's name. The following paragraphs contain the explanation and a word of justification that allows it to stand on its own.

The convexity constraint (10) produces a scaling in the $\mathbf{y}$-variables which is echoed in the objective function. As a matter of fact, problem (8)–(11) no longer belongs to the family of network problems. Nevertheless, that scaling does not compromise the existence of a cycle in $G(\mathbf{x}^0)$, but it does create a distortion of the cost associated with said cycle. The meaning of this distortion resides in the fact that (8)–(11) finds a single directed cycle with the smallest average cost, the average being taken over the number of arcs (or nodes) in the cycle. Notice the use of the word cycle against formulation (8)–(11) which we have explicitly excluded from the network family. The concept is so important, we take the time to break the flow of the text to carry an explanation.

Define $W^0 := \{(i,j) \in A(\mathbf{x}^0) \mid y^0_{ij} > 0\}$ as the set of active variables in an optimal solution $\mathbf{y}^0$ to formulation (8)–(11). Granted W^0 describes a single cycle, constraint set (9) guarantees that the value is the same for all the variables that are actually present in that selected cycle. Therefore, we must have $y^0_{ij} = 1/|W^0|, \ \forall (i,j) \in W^0$. Furthermore, we say W^0 is directed with respect to the orientation of the arcs in $G(\mathbf{x}^0)$ corresponding to the selected positive variables in $\mathbf{y}^0$. While the notation is abusive, the burden of an additional variable for values so closely related is not worthwhile. In any single cycle, at most one of y^0_{ij} or y^0_{ji} may be positive which in turn satisfies the flow condition $y_{ij}y_{ji} = 0$. Fortunately, the expectancy of this particular kind of solution is not a strong restriction as it is synonymous of an extreme point solution of the linear program (8)–(11). There is one notable exception to this one-way rule which can only happen when $\mathbf{x}^0$ is optimal. Since the identified cycle can be discarded for lack of improvement, so can the exception. From now on, a solution to the pricing step is assumed to honor the design of the minimum mean cycle problem meaning that W^0 is a single directed cycle. Furthermore, we can interchangeably speak of cycle W^0 on $G(\mathbf{x}^0)$ or $A(\mathbf{x}^0)$. Finally, note that in the dual formulation, one can additionally impose $\mu \leq 0$. As a consequence, the associated convexity constraint in (8)–(11) becomes a less than or equal inequality and the primal pricing problem is always feasible even if the residual network is acyclic (in which case $\mu = 0$). Observe that if the obtained cycle has a negative mean reduced cost, $\mathbf{x}^0$ is not optimal for (1). The solution of the pricing step can therefore be seen as a *direction*. By definition of the residual network, it even qualifies as a strictly improving direction.

Remark. The justification for the validity of the primal version of the pricing step might go as follows. We are looking to improve current solution $\mathbf{x}^0$ using the concept of negative cost cycles. More specifically, we are looking for the existence of such a cycle, say W^0 on $G(\mathbf{x}^0)$. Observe that the residual capacities are not relevant in the cycle identification process. Then again, omitting these quantities from formulation (2)–(4), that is, removing $y_{ij} \leq r^0_{ij}, \ \forall (i,j) \in A(\mathbf{x}^0)$, creates an unbounded circulation problem. The silver lining comes from the realization that this new problem is a cone for which the unique extreme point solution $\mathbf{y} = \mathbf{0}$ reflects *status quo*. Non linear optimization has an impressive inventory of choices to search among improving directions. These choices all have their rationalization but ultimately mean that the cone is cut according to some metric. One of the recognized choices is the convexity constraint imposed on the selection. Indeed, for any non-zero solution in the cone, there exists a scaled one such that $\mathbf{1}^\intercal \mathbf{y} = 1$. Historically speaking, the pricing step is reported as an abstract form of its interpretation, that is, $\min_W \sum_{(i,j)\in W} d_{ij}/|W|$. Whether Goldberg and Tarjan [13] accidentally built the minimum mean cycle-canceling algorithm according to these principles or it was meticulously devised is unclear, the conclusion is all the same: the convexity constraint is indeed enlisted in (8)–(11).

2.3. Algorithmic process

MMCC is initialized with a feasible flow $\mathbf{x}^0$. At every iteration $k \geq 0$, the pricing step solution $\mathbf{y}^k$ identifies a minimum mean cost cycle $W^k := \{(i,j) \in A(\mathbf{x}^k) \mid y_{ij}^k > 0\}$ taking value μ^k in $G(\mathbf{x}^k)$. Flow units are sent along this cycle according to a control mechanism which relies solely on the residual capacities $\mathbf{r}^k$. A new solution $\mathbf{x}^{k+1}$ is obtained and $G(\mathbf{x}^{k+1})$ is updated accordingly. This process is repeated until the residual network contains no negative cycle.

Let us take a look at some computations that can be done regarding the transition between two iterations. The flow of every arc in the negative cost cycle W^k can be augmented by the smallest residual capacity on the cycle $\delta^k := \min_{(i,j)\in W^k} r_{ij}^k$, hence the new solution becomes

$$x_{ij}^{k+1} = x_{ij}^k + \delta^k |W^k| (y_{ij}^k - y_{ji}^k), \quad \forall (i,j) \in A, \tag{12}$$

and the improvement Δz^k of the objective function in (1) is given by

$$\Delta z^k := \delta^k |W^k| \mu^k = \delta^k \sum_{(i,j)\in W^k} d_{ij} < 0. \tag{13}$$

Notice that both δ^k and Δz^k evaluate to integers if certain conditions are verified. The former requires the integrality of the bounds as well as the demands/supplies while the latter depends on the integrality of the costs.

It stands to reason that MMCC is already in the works with Ford and Fulkerson [9] providing the concept of *augmenting paths* between solutions while Edmonds and Karp [7] show that a particular selection of augmenting paths would be more efficient on the maximum flow problem. While we use the latter to present an application of MMCC, the reader is invited to appreciate the narrative description as a tribute to the aforementioned papers. The illustrative example also serves to get a feel for the subsequent complexity analysis.

2.4. Illustrative example: the maximum flow problem

The maximum flow problem is a particular instance of network optimization in which a source s and a sink t are connected through a capacitated subnetwork. The goal is to maximize the outgoing flow of the source under the restriction of the usual flow conservation constraints. One should realize the null cost structure of all the arcs except $x_{ts} \geq 0$ for which $c_{ts} = -1$. Let us apply MMCC and assume lower bounds are null for all arcs, meaning that $\mathbf{x}^0 = \mathbf{0}$ is feasible.

It is worthwhile to notice that the cycle found on the residual network at any iteration $k \geq 0$ is constructed in two parts: a path from s to t and the lone variable y_{ts}. Let W^k be the negative cycle identified at iteration k, hence $\mu^k = -1/|W^k|$. The pricing step sequentially favors the smallest paths (in number of arcs) from s to t starting from length 1 to $n-1$ until optimality is reached. The sequence of μ^k is non-decreasing and takes its values from the finite set $\left\{\frac{-1}{2}, \frac{-1}{3}, \frac{-1}{4}, \ldots, \frac{-1}{n}\right\}$. When the path length changes from 2 to a longer one, the increase factor of μ takes a value among range $\{\frac{2}{3}, \frac{2}{4}, \ldots, \frac{2}{n}\}$. Observe that higher values of the increase factor induce smaller jumps on μ. Therefore, the minimal increase factors $\left\{\frac{2}{3}, \frac{3}{4}, \ldots, \frac{n-1}{n}\right\}$ of μ^k, for each length level, are computed using adjacent values of μ, the smallest possible one being attained when going from level $\frac{-1}{n-1}$ to $\frac{-1}{n}$ and measured by $(1 - 1/n)$. Fig. 2 depicts the behavior of μ^k, $k \geq 0$, on a network comprising 502 nodes and 10,007 arcs: only four levels of μ are required within 188 iterations at which point $\mu = 0$ thus proving optimality and discarding the associated cycle defined by variables $y_{st}^{188} = y_{ts}^{188} = 1/2$. Observe that in this particular example, the increase factors are $\frac{5}{6}$, $\frac{6}{7}$ and $\frac{7}{8}$ all of which producing a bigger increase on μ than would have $\frac{501}{502}$.

We draw the reader's attention on solving the pricing step. We are looking, on the residual network, for the shortest path (in number of arcs) from s to t. This can be done in $O(m)$ using a breadth-first-search algorithm. Next, we derive from the previous paragraph that there are at most $n-1$ increases of μ, and since every iteration identifies a path through which at least one arc is saturated with the step size, each increase is attained within m iterations, hence $O(mn)$ iterations. In total, Goldberg and Tarjan [13] realize that applying MMCC to the maximum flow problem exactly corresponds to the strongly polynomial algorithm of Edmonds and Karp [7] which runs in $O(m^2 n)$ time.

Please cite this article in press as: J.B. Gauthier, et al., About the minimum mean cycle-canceling algorithm, Discrete Applied Mathematics (2014), http://dx.doi.org/10.1016/j.dam.2014.07.005

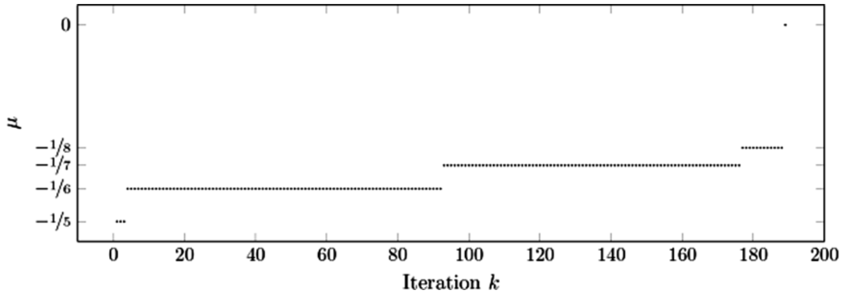

Fig. 2. Smallest reduced cost μ for a maximum flow problem.

3. Complexity analysis

The complexity analysis is implicitly decomposed in two parts: the *outer loop* and the *bottleneck*. Obtaining the global runtime is then a matter of factoring out these complexities. In MMCC, the natural definition of the bottleneck relates to the pricing step and we therefore study the upper bound on the number of calls made to the latter. For the sake of argument, one can think of the bottleneck as a group of calls which, in the end, is purely cosmetic. Yet, an efficacy gain is made if solving for a group can be done more efficiently than would the sequential operations. That is where lies the significance of the so-called *phases*.

Although this paper recruits its inspiration from the very fine presentation of the minimum mean cycle-canceling algorithm proposed by Goldberg and Tarjan [13] and described in [1], the presentation is reorganized to first thoroughly discuss the outer loop analysis (Sections 3.1–3.4) and then spend time on the bottleneck management (Section 3.5). We also rapidly divert to phase-wise results in accordance with our understanding of the Cancel-and-Tighten strategy. The latter is presented in the bottleneck management section opposite the traditional pricing problem. Ultimately, we consolidate all complexity results in the summary (Section 3.6). The latter also points out different practical aspects of the algorithm by using computational results.

We start with basic properties of the algorithm which lead to the actual complexity analysis. The first proposal of Goldberg and Tarjan [13] is a weakly polynomial behavior of $O(mn\log(nC))$ iterations for integer arc costs while the second establishes the strongly polynomial result of $O(m^2n\log n)$ iterations for arbitrary real-valued arc costs. Radzik and Goldberg [17] refine it to $O(m^2n)$ iterations and also show this bound to be tight. The concept of phase is strategically positioned after the first complexity result expressed in terms of iterations to allow the reader to appreciate the similarity. The bottleneck management brings the Cancel-and-Tighten strategy into play and reduces the global runtime complexity to $O(m^2n\log n)$, our new complexity result for the minimum mean cycle-canceling algorithm.

3.1. In embryo

Let us recall a fundamental network flow property before dwelling in the algorithmic analysis. It examines the relationship with the cost and the reduced cost of a cycle.

Cycle cost. For any vector $\boldsymbol{\pi}$ of node potentials, the cost and the reduced cost of a directed cycle W in $G(\mathbf{x}^k)$, $k \geq 0$, are equal. Finding a minimum mean cost cycle W^k in $G(\mathbf{x}^k)$ is therefore equivalent to finding a minimum mean *reduced* cost cycle W^k in $G(\mathbf{x}^k)$. Hence, the optimal value of the objective function in pricing problem (8)–(11) computes

$$\mu^k = \sum_{(i,j)\in W^k} d_{ij}y_{ij}^k = \sum_{(i,j)\in W^k} \frac{d_{ij}}{|W^k|} = \sum_{(i,j)\in W^k} \frac{\bar{d}_{ij}}{|W^k|}. \tag{14}$$

The first equality sums over the optimal cycle, the second uses the fact that we know all strictly positive $\mathbf{y}$-variables are equal to one another, and the last recalls the equivalence between the cost and the reduced cost of a cycle.

Optimality parameter μ. The mechanics of the minimum mean cycle-canceling algorithm do not require the use or computation of reduced costs. Indeed, MMCC relies solely on the primal optimality condition to achieve optimality. The complexity analysis however exploits the dual and complementary slackness conditions using the equivalences provided by (14). Ultimately, the idea is to study the convergence towards zero of μ^k, the current most negative reduced cost. The synonymy is granted as a side effect of Proposition 3, for which the proof is given using linear programming tools, and is the reason we interchangeably use the expression *optimality parameter*. With respect to $\boldsymbol{\pi}^k := [\pi_i^k]_{i\in N}$ at iteration $k \geq 0$, let $\bar{c}_{ij}^k := c_{ij} - \pi_i^k + \pi_j^k$, $(i,j) \in A$, be the reduced cost of variable x_{ij}. In the same way, $\bar{d}_{ij}^k := d_{ij} - \pi_i^k + \pi_j^k$ is the reduced cost of variable y_{ij}, $(i,j) \in A(\mathbf{x}^k)$. Observe that the superscript is understood to mean the computation is done with the corresponding vector $\boldsymbol{\pi}^k$ of node potentials.

Please cite this article in press as: J.B. Gauthier, et al., About the minimum mean cycle-canceling algorithm, Discrete Applied Mathematics (2014), http://dx.doi.org/10.1016/j.dam.2014.07.005

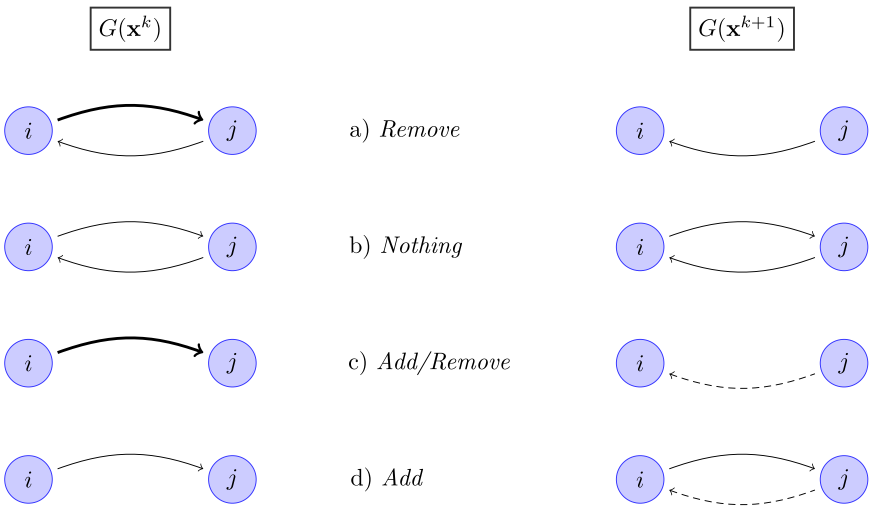

Fig. 3. Aftermath of cycle-canceling in the residual network.

Proposition 3 (*Goldberg and Tarjan [13, Theorem 3.3]*). *Given a non-optimal solution $\mathbf{x}^k$, $k \geq 0$, there exists some vector $\boldsymbol{\pi}^k$ such that the optimality parameter is equal to the most negative reduced cost, i.e., $\mu^k = \min_{(i,j)\in A(\mathbf{x}^k)} \bar{d}_{ij}^k$. Moreover all arcs of the identified cycle W^k share that same value, i.e., $\bar{d}_{ij}^k = \mu^k, \ \forall (i,j) \in W^k$.*

Proof. At iteration k, constraint set (7) can be written as $\mu \leq d_{ij} - \pi_i + \pi_j, \forall (i,j) \in A(\mathbf{x}^k)$. At optimality, $\mu^k \leq \bar{d}_{ij}^k, \ \forall (i,j) \in A(\mathbf{x}^k)$ and the objective function (6) pushes μ^k to the smallest reduced cost. Furthermore, the complementary slackness conditions guarantee that the equality holds in (7) for all $y_{ij}^k > 0$, that is, $\mu^k = \bar{d}_{ij}^k, \ \forall (i,j) \in W^k$. □

ϵ-optimality conditions. The complexity analysis is born out of a chain of arguments that is bound by a series of equivalences. Many strongly polynomial algorithms for CMCF use the concept of ϵ-optimality obtained by relaxing the complementary slackness constraints, see for example, [2,19,21,11,13,17]. Ultimately, it turns out that parameters ϵ and μ are linked by an equality expression, that is, $\epsilon = -\mu$. We argue that it all comes together with the linear programming formulation of the pricing problem's dual. This line of thoughts allows us to discard the ϵ-parameter and is indeed the reason we cannot say with certainty it was understood as such. In the spirit of the coined expression, a μ-optimal solution $\mathbf{x}^k$ can be understood in the same way as its ϵ-counterpart, that is, relaxed complementary slackness conditions which provide approximate optimality, see [1, relations (10.1)–(10.2)]. Feel free to compare (5) with the following relaxed conditions:

$$x_{ij}^k = \ell_{ij} \quad \text{if } \bar{c}_{ij} > -\mu; \qquad x_{ij}^k = u_{ij} \quad \text{if } \bar{c}_{ij} < \mu; \qquad \ell_{ij} \leq x_{ij}^k \leq u_{ij} \quad \text{if } \mu \leq \bar{c}_{ij} \leq -\mu. \tag{15}$$

It should come as no surprise that solution $\mathbf{x}^k$ is μ^k-optimal. The reader may want to verify that the equivalent condition on the residual network $G(\mathbf{x}^k)$ questions whether there exists $\boldsymbol{\pi}$ such that $\bar{d}_{ij} \geq \mu, \ \forall (i,j) \in A(\mathbf{x}^k)$.

The following propositions stand outside the scope of complexity theorems for several reasons. The first is that they are very strong results for MMCC. The second is that their validity is independent of any assumptions regarding problem data. The third is that we ascertain the comprehension of the transitive mechanics between two solutions.

Proposition 4 (*Goldberg and Tarjan [13, Lemma 3.5]*). *For any two consecutive iterations k and $k+1$, the value of μ is non-decreasing, that is, $\mu^k \leq \mu^{k+1}$, $k \geq 0$.*

Proof. The proof consists of examining the effect of canceling cycle W^k in light of the new solution $\mathbf{x}^{k+1}$ and more specifically the marginal modifications incurred in $G(\mathbf{x}^{k+1})$. There are only four possibilities as displayed in Fig. 3. First off, either the residual network $G(\mathbf{x}^k)$ contains arcs in both directions or only one between nodes i and j. Secondly, either the flow that passes on an arc of cycle W^k saturates it (arcs in bold) or not.

By Proposition 3, vector $\boldsymbol{\pi}^k$ ensures that $\bar{d}_{ij}^k \geq \mu^k$ in $G(\mathbf{x}^k)$ such that $\bar{d}_{ij}^k = \mu^k, \ \forall (i,j) \in W^k$. In $G(\mathbf{x}^{k+1})$, the saturated arcs in cycle W^k are removed and new arcs appear in the reverse direction with a reduced cost equal to $-\mu^k > 0$. Therefore, by construction, every arc of $G(\mathbf{x}^{k+1})$ has a reduced cost $\bar{d}_{ij}^k \geq \mu^k$ computed with respect to $\boldsymbol{\pi}^k$. Since the mean cost of a cycle is at least as great as the minimum cost of any of its terms, $\mu^{k+1} \geq \mu^k$. □

We take the time to stress the fact that MMCC is an iterative algorithm. Even though the previous proposition is true of any two consecutive iterations, the proof still uses a point of reference. The following proofs base their arguments on a sequence of iterations and it is imperative to take a step back to appreciate the global picture.

Proposition 4 is not sufficient to provide convergence properties. It is indeed mandatory for μ to strictly increase sporadically towards zero. Define a *jump* on the optimality parameter as the situation where there exists some factor $0 \leq \tau < 1$ such that $\mu^{k+1} \geq \tau\mu^k > \mu^k$, for $k \geq 0$. Recall Fig. 2 which exhibits this behavior for the maximum flow problem.

Proposition 5 (*Goldberg and Tarjan [13, Lemma 3.6]*)**.** *Given a non-optimal solution* $\mathbf{x}^k,\ k \geq 0$*, a sequence of no more than m iterations allows* μ *to jump by a factor of at least* $(1 - {}^1/_n)$.

Proof. Recall that the cost and the reduced cost of a cycle take the same value. Hence, the reduced costs can be computed with any set of potentials. In order to show the statement, we use vector $\boldsymbol{\pi}^0$ found at iteration 0, hence $\bar{d}_{ij}^0 \geq \mu^0$, $\forall(i,j) \in A(\mathbf{x}^0)$. At iteration $k \geq 1$, we distinguish two types of cycles according to the following definitions. A cycle W^k of Type 1 contains only arcs of strictly negative reduced costs, i.e., $\bar{d}_{ij}^0 < 0, \forall(i,j) \in W^k$, while a cycle of Type 2 contains at least one arc with a non-negative reduced cost, i.e., $\exists(i,j) \in W^k \mid \bar{d}_{ij}^0 \geq 0$.

We prove that within m iterations, the algorithm finds a Type 2 cycle otherwise the optimal solution to (1) has been reached. The same reasoning used in Proposition 4 allows us to realize two things regarding Type 1 cycles. First, there is at least one saturated arc of strictly negative reduced cost that is removed in the next residual network. Second, reversed arcs added all have strictly positive reduced costs with respect to $\boldsymbol{\pi}^0$. Since there are no more than m arcs with strictly negative costs, optimality is reached after at most m consecutive Type 1 cancelations.

For $l \leq m$, assume a Type 2 cycle W^l is found, then at least one of its arcs has a non-negative reduced cost. The worst case scenario in terms of mean reduced cost is to pass through $|W^l| - 1 \leq n - 1$ arcs of cost μ^0 and one of zero. As such, $\mu^l \geq \frac{(|W^l|-1)\,\mu^0}{|W^l|} \geq \frac{(n-1)\,\mu^0}{n} = \left(1 - \frac{1}{n}\right)\,\mu^0$. □

Proposition 6. *Given a non-optimal solution* $\mathbf{x}^k,\ k \geq 0$*, a sequence of no more than mn iterations allows* μ *to jump by a factor of at least* ${}^1/_2$.

Proof. Basic calculus shows that $(1 - {}^1/_n)^n < {}^1/_2,\ \forall n \geq 2$ as it converges to ${}^1/_e$. This means that every mn iterations, the value of μ increases by a factor of at least ${}^1/_2$. □

3.2. *Integer costs:* $O(n\log(nC))$ *phases*

This section contains the first installment regarding the actual complexity of MMCC. The only assumption is that all cost data are integers. Recall that $C := \max_{(i,j)\in A} |c_{ij}|$.

Theorem 1 (*[13, Theorem 3.7]*)**.** *Given a capacitated network with integer arc costs, MMCC performs* $O(mn\log(nC))$ *iterations.*

Proof. Let us consider the values of μ obtained for each iteration as a sequence. This sequence may be bounded below because no mean cycle can cost less than $-C$. It is also bounded above by the highest strictly negative reduced cost cycle computed as $\frac{(-1)+(n-1)0}{n} = -{}^1/_n$. In light of Proposition 6, it is possible to construct a geometric progression of reason at least ${}^1/_2$ with some elements of the sequence $\{\mu^k\}$. As it stands, solving for the power of $-C\left(\frac{1}{2}\right)^\kappa = -{}^1/_n$, the geometric system that prevails every mn iterations, one obtains $\kappa = \log(nC)$. Therefore, the image of the objective function can always be traversed in $O(mn\log(nC))$ iterations. □

Let us revisit Proposition 5 in order to extract what, in retrospective, seems like a key property. Indeed, Goldberg and Tarjan [13] use a Type 2 cancellation as a marker for the jump factor on the optimality parameter μ. We stress that while it is an elegant (read sufficient) condition, it is certainly not necessary. Type 1 cancellations can indeed run into jumps however we still do not know how to measure them. Type 2 cancellations are thus markers for *measurable* jumps. It is probably what prompts Radzik and Goldberg [17] to define a phase with a definition that is much closer to the spirit of Proposition 5.

Definition 1. A *phase* is a sequence of iterations terminated by a Type 2 cycle. A *phase solution* $\mathbf{x}^h,\ h \geq 0$, is understood as the solution at the beginning of phase h.

We stress that while the two phase numbers h and $h+1$ are consecutive, by Proposition 5, the number of cycles canceled within phase h is at most m. Let $l \leq m$ be that length. For the remainder of this article, the notations k and h are respectively reserved for iteration and phase based operations. The notation l is used to refer to the last cycle identified in the phase, that is, the Type 2 cycle.

Proposition 7. *Given a non-optimal phase solution* $\mathbf{x}^h,\ h \geq 0$*, the optimality parameter obtained on the following phase solution strictly increases by a factor of least* $(1 - {}^1/_n)$ *and increases by a factor of at least* ${}^1/_2$ *every n phases.*

Proof. The proof is immediate from the definition of a phase. Indeed, a Type 2 cycle implies a measurable jump on μ and means that the relation of the optimality parameter between two consecutive phases is $\mu^{h+1} \geq (1 - {}^1/_n)\,\mu^h > \mu^h$. Moreover, $\mu^{h+n} \geq (1 - {}^1/_n)^n\,\mu^h > \frac{1}{2}\,\mu^h$. □

Assuming a mechanism that allows the resolution to terminate a phase under the same conditions, we can make abstraction of the Type 1 iterations and express the outer loop in terms of phases instead of iterations. For instance, the weakly polynomial complexity of $O(mn\log(nC))$ iterations obtained in Theorem 1 can be rewritten as $O(n\log(nC))$ phases. While one might argue that this change is purely cosmetic, Section 3.5 holds the key to the justification. In a nut shell, the implementation choice dictates the actual complexity depending on whether a phase is solved in an integrated manner or not.

Although the following idea is already present in the proof of Proposition 5, we feel it warrants a repetition. The same node potentials $\boldsymbol{\pi}^h$ can be used throughout the whole phase. As such, in the adaptation to a phase-wise analysis, the reduced costs of each arc is also unchanged during the phase. Consider a sequence of iterations consisting of consecutive Type 1 cycles starting from phase solution $\mathbf{x}^h$. Solving the pricing step at every iteration implicitly associates the minimum mean cycle found with a tight vector of node potentials along with the optimality parameter in accordance with the primal–dual formulations. However, by definition the very first vector of node potentials, $\boldsymbol{\pi}^h$, validates the Type 1 condition of the cancellations throughout the series of residual networks traversed. In other words, while $\boldsymbol{\pi}^h$ might not be the tightest vector of node potentials for every residual networks associated with this sequence, it is sufficient to allow the identification of these cycles. Let it be clear that the existence of this sustainable vector $\boldsymbol{\pi}^h$, valid until a new vector $\boldsymbol{\pi}^{h+1}$ is required, is irrelevant to the solving process of MMCC. What is important to retain is that the node potentials can have a lasting effect of at most m iterations, the maximal length of a phase during the resolution process.

3.3. Arbitrary costs: $O(mn\log n)$ phases

So far, we have shown that MMCC runs in weakly polynomial time from two different perspectives: the strictly decreasing objective function and the sequence of strictly increasing μ^h which can, in some way, be interpreted as a subsequence of $\{\mu^k\}$. In order to speak of strongly polynomial time, a new angle is required. That angle is named *arc fixing*. The idea is to tag an arc as being fixed at one of its bounds. In line with the complementary slackness optimality conditions (5), this tagging occurs when the reduced cost associated with an arc is sufficiently far from zero, that is, positively large enough for fixing a variable at its lower bound or negatively small enough for fixing it at its upper bound.

The following proposition states the arc fixing rule. The argument is based on the logical implication of the complementary slackness conditions (5) to create a correspondence between the current value of the optimality parameter μ^k and the flow value of certain arcs.

Proposition 8 (*[13, Theorem 3.8]*)**.** *Let $k \geq 0$ denote a non-optimal iteration number. Arc fixing occurs for arc $(i,j) \in A$ if and when $|\bar{c}^k_{ij}| \geq -2n\mu^k$. Expressed in terms of $G(\mathbf{x}^k)$, we have: arc $(i,j) \in A(\mathbf{x}^k)$ is fixed at zero if and when $\bar{d}^k_{ij} \geq -2n\mu^k$.*

Proof. The proof establishes a contradiction with the previous properties when a fixed arc is used. Assume, without loss of generality, that arc $(i,j) \in A$ has reached $\bar{c}^k_{ij} \geq -2n\mu^k$ at iteration k. According to Proposition 3, $A(\mathbf{x}^k)$ must contain arc (i,j) and not (j,i) because its reduced cost would be less than μ^k, which means that $x^k_{ij} = \ell_{ij}$ and $\bar{d}^k_{ij} = \bar{c}^k_{ij} \geq -2n\mu^k$.

Assume $\mathbf{x}^s$, $s > k$, values $x^s_{ij} > 0$. By Proposition 2, $\mathbf{x}^s$ equals $\mathbf{x}^k$ plus the flow on at most m directed cycles in $G(\mathbf{x}^k)$, and *vice versa*. Hence, there exists a cycle in $G(\mathbf{x}^k)$, say W^+, using arc (i,j). The mean reduced cost of this cycle, denoted $\mu(W^+)$, could even be valued by

$$\mu(W^+) \geq \frac{-2n\mu^k + (|W^+|-1)\ \mu^k}{|W^+|} \geq \frac{-2n\mu^k + (n-1)\ \mu^k}{|W^+|} = \frac{-(n+1)}{|W^+|}\mu^k.$$

The existence of cycle W^+ in $G(\mathbf{x}^k)$ means that the reverse cycle denoted W^- exists in $G(\mathbf{x}^s)$ with a mean reduced cost of $\mu(W^-) = -\mu(W^+) \leq \frac{n+1}{|W^+|}\mu^k < \mu^k$, which is a contradiction with Proposition 4 on the fact that the optimality parameter μ is non-decreasing.

The correspondence between the original and the residual arcs is subtle. Arc $(i,j) \in A$ is fixed at ℓ_{ij} because $\bar{d}^k_{ij} \geq -2n\mu^k$ such that arc $(i,j) \in A(\mathbf{x}^s)$, $s > k$ is fixed at zero. The proof for arc $(i,j) \in A$ being fixed at its upper bound u_{ij} when $\bar{c}^k_{ij} \leq 2n\mu^k$ or equivalently for arc $(j,i) \in A(\mathbf{x}^k)$ being fixed at 0 when $\bar{d}^k_{ji} \geq -2n\mu^k$ is similar. □

This proof assumes the non-decreasing property of the optimality parameter over the iterations which is lost when using the Cancel-and-Tighten strategy. A careful yet straightforward adaptation can be made using the following proposition.

Proposition 9. *Let $h \geq 0$ denote a non-optimal phase number. Arc fixing occurs for arc $(i,j) \in A$ if and when $|\bar{c}^h_{ij}| \geq -2n\mu^h$. Expressed in terms of $G(\mathbf{x}^h)$, we have: arc $(i,j) \in A(\mathbf{x}^h)$ is fixed at zero if and when $\bar{d}^h_{ij} \geq -2n\mu^h$.*

This proposition holds exactly the same value for the complexity analysis because the second installment of the theoretical complexity only uses phases to capture the concept of arc fixing. Let us define a *block* accordingly.

Definition 2. A *block* is a sequence of phases terminated by the fixing of at least one arc. A *block solution* is understood as the solution at the beginning of a block.

Please cite this article in press as: J.B. Gauthier, et al., About the minimum mean cycle-canceling algorithm, Discrete Applied Mathematics (2014), http://dx.doi.org/10.1016/j.dam.2014.07.005

Theorem 2 ([13, Theorem 3.9]). *Given a capacitated network with arbitrary real-valued arc costs, MMCC performs $O(mn \log n)$ phases.*

Proof. This proof relies on the concept of arc fixing. The idea is to show that at least one new arc is fixed within a limited number of phases. We show this bound to be $n(\lceil \log n \rceil + 1) \equiv O(n \log n)$ phases. As such, consider this particular sequence of phases as a block.

For phase number h, let $\mathbf{x}^\circ = \mathbf{x}^h$ and $\mathbf{x}^\bullet = \mathbf{x}^{h+n(\lceil \log n \rceil + 1)}$ be respectively the solutions prior to the first and after the last iteration of any given block. By Proposition 7, we know μ increases by a factor of at least 1/2 every n phases, so the increase in a block is

$$\mu^\bullet = \mu^{\circ + n(\lceil \log n \rceil + 1)} \geq \frac{1}{2^{\lceil \log n \rceil + 1}} \mu^\circ \geq \frac{1}{2n} \mu^\circ.$$

Found on $G(\mathbf{x}^\circ)$, consider cycle W° with mean reduced cost μ°, a value independent of the potentials used. Therefore, with respect to $\boldsymbol{\pi}^\bullet$, the arc reduced costs in W° cannot all be greater than μ°. Hence, there exists a variable, say $y_{ji},\ (j, i) \in W^\circ$, with a reduced cost $\bar{d}_{ji}^\bullet$ at most equal to μ°, that is, $\bar{d}_{ji}^\bullet \leq \mu^\circ \leq 2n\mu^\bullet$. On $G(\mathbf{x}^\bullet)$, variable y_{ij} appears with a reduced cost of $\bar{d}_{ij}^\bullet = -\bar{d}_{ji}^\bullet \geq -2n\mu^\bullet$. By Proposition 8, the value of variable y_{ij} does not change anymore and the corresponding variable x_{ij} is fixed at its lower bound. Moreover, as part of the optimal cycle W°, the algorithm modifies the value of x_{ij} in the very first iteration of the block. In retrospective, arc $(j, i) \in G(\mathbf{x}^\circ)$ must have saturated the residual capacity and it is quite interesting to note that the confirmation of the flow value comes later than the time at which it is established. The proof for arc fixing at upper bound is a straightforward adaptation.

Let it be clear that $\mathbf{x}^\bullet$ is a block solution in that it becomes $\mathbf{x}^\circ$ in the following block. All in all, each block fixes a different arc. Since there are m arcs, the proposed complexity is achieved. □

3.4. Arbitrary costs: $O(mn)$ phases

The final piece of the complexity puzzle comes much later than the seminal paper of Goldberg and Tarjan [13] which elaborates all the propositions we have seen thus far. Although they utilize properties of the complementary slackness conditions (5) at every iteration, Radzik and Goldberg [17] make use of the properties of an optimal primal–dual pair in their complexity analysis, establishing the best possible strongly polynomial result for the minimum mean cycle-canceling algorithm.

Radzik and Goldberg [17] obviously have such a good understanding of MMCC that it really allows them to think outside the box. While an optimal vector of node potentials $\boldsymbol{\pi}^*$ is unknown, we can use the fact that it does exist. Denote the reduced costs computed using such a set of optimal potentials by $\bar{c}_{ij}^* := c_{ij} - \pi_i^* + \pi_j^*,\ \forall (i, j) \in A$.

Proposition 10 ([17, Lemma 9]). *Let $k \geq 0$ denote a non-optimal iteration number. Implicit arc fixing occurs for arc $(i, j) \in A$ if and when $|\bar{c}_{ij}^*| > -n\mu^k$. Expressed in terms of $G(\mathbf{x}^k)$, we have: arc $(i, j) \in A(\mathbf{x}^k)$ is implicitly fixed at zero if and when $\bar{d}_{ij}^* > -n\mu^k$.*

Proof. If $\bar{c}_{ij}^* > -n\mu^k$, then $\bar{c}_{ij}^* > 0$ and $x_{ij}^* = \ell_{ij}$ by the complementary slackness optimality conditions (5). Now assume arc $(i, j) \in A$ has reached $\bar{c}_{ij}^* > -n\mu^k$ at iteration k but $x_{ij}^k > \ell_{ij}$. By Proposition 2, $\mathbf{x}^*$ equals $\mathbf{x}^k$ plus the flow on at most m directed cycles in $G(\mathbf{x}^k)$, and *vice versa*. Hence there exists a cycle W^+ in $G(\mathbf{x}^k)$ using variable y_{ji} to push the flow back towards ℓ_{ij}. The reverse cycle W^- exists in $G(\mathbf{x}^*)$ using arc (i, j) with $\bar{d}_{ij}^* = \bar{c}_{ij}^* > -n\mu^k$. Because optimal arc reduced costs on $G(\mathbf{x}^*)$ are greater than or equal to zero, $\mu(W^-) > \frac{-n\mu^k}{|W^-|}$. Therefore $\mu(W^+) = -\mu(W^-) < \frac{n\mu^k}{|W^-|} < \mu^k$, a contradiction on the optimality of μ^k at iteration k. The proof for an arc $(i, j) \in A$ being implicitly fixed at its upper bound u_{ij} when $\bar{c}_{ij}^* < n\mu^k$ is similar. □

Once again, the following analysis still revolves around phases. Using the Cancel-and-Tighten strategy modifies the statement of the previous proposition without compromising its value. The comparison must be done against the optimality parameter computed at each phase h.

Proposition 11. *Let $h \geq 0$ denote a non-optimal phase number. Implicit arc fixing occurs for arc $(i, j) \in A$ if and when $|\bar{c}_{ij}^*| > -n\mu^h$. Expressed in terms of $G(\mathbf{x}^h)$, we have: arc $(i, j) \in A(\mathbf{x}^h)$ is implicitly fixed at zero if and when $\bar{d}_{ij}^* > -n\mu^h$.*

The third and last installment of the complexity analysis is at hand. The avid reader might even recall the introductory complexity proposition. With a statement so closely matching that of Theorem 2, the expectation of an analogous proof is annihilated from the start. While the previous proof bounds the length of each block against a uniquely defined value, the following propositions show that the more subtle kind of implicit arc fixing of Propositions 10 and 11 happens in a unpredictable manner. Nevertheless, Radzik and Goldberg [17] are able to prove by a global analysis that the whole implicit arc fixing process is itself bounded. The first step towards this result is to bind the number of phases to the number of arcs contained in the cycles identified within these phases. The latter in turn grants a tighter measurable jump factor for the optimality parameter according to the following observation.

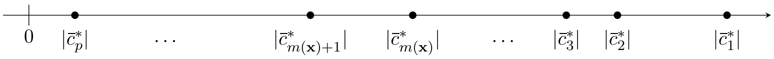

Fig. 4. Optimal absolute arc reduced costs on the axis.

Tighter jump factor. Given a non-optimal phase solution $\mathbf{x}^h,\ h \geq 0$, the impact of the jump factor for a phase of length $l \leq m$ can be tightened by the size of the Type 2 cycle W^l. This is evaluated by

$$\mu^{h+1} \geq \left(1 - \frac{1}{|W^l|}\right)\mu^h \geq \left(1 - \frac{1}{n}\right)\mu^h. \tag{16}$$

The proof of Proposition 5 can be trivially modified to accommodate this observation by using, for the Type 2 cycle, $|W^l| - 1$ instead of $n - 1$. Yet, this new tighter jump factor strongly endorses the empirical behavior of cycle sizes much smaller than n. The parameter, $L(\mathbf{x}^h)$, used to bind these values together emerges from the following construction.

Let $e \in A$ denote an arc of network G with optimal reduced cost $\bar{c}^*_e$. For the purpose of the complexity analysis, the arcs are sorted in decreasing order of their *absolute optimal reduced cost* values, that is, $0 \leq |\bar{c}^*_m| \leq \cdots \leq |\bar{c}^*_2| \leq |\bar{c}^*_1|$. While the notation is introduced hereafter, the rightmost part of Fig. 4 visually explains the sort.

For a given solution $\mathbf{x}$, let $m(\mathbf{x}) := \min\{e \in A : |\bar{c}^*_e| \leq -n\mu(\mathbf{x})\}$ be the smallest index for which arcs $e \geq m(\mathbf{x})$ have not yet been implicitly fixed and $S_{m(\mathbf{x})} := \sum_{e=m(\mathbf{x})}^{m} |\bar{c}^*_e|$ be the sum of their absolute reduced costs.

As optimality parameter μ increases towards zero, $|\bar{c}^*_1| > -n\mu$ and arc $e = 1$ is the first arc implicitly fixed. Next implicit fixing occurs for arc $e = 2$, and so forth for the remaining arcs. Arc variable x_e with an index value $e < m(\mathbf{x})$ cannot be part of an improving cycle for a non-optimal solution $\mathbf{x}$ because, by Proposition 10, its value does not change anymore. Therefore, index $m(\mathbf{x})$ increases towards m whereas largest absolute reduced cost $|\bar{c}^*_{m(\mathbf{x})}|$ and sum $S_{m(\mathbf{x})}$ decrease towards zero. When $S_{m(\mathbf{x})} = 0$, every variable with an optimal reduced cost different from zero has been implicitly fixed while any free variable lies within its interval domain with a zero-value optimal reduced cost, hence satisfying the necessary and sufficient complementary slackness optimality conditions (5). As such, let $p \leq m$ denote the largest arc index for which the optimal absolute reduced cost is strictly positive. The latter is the last arc variable which can answer to the implicit arc fixing rule. Observe that this means that $S_{m(\mathbf{x})} := \sum_{e=m(\mathbf{x})}^{m} |\bar{c}^*_e| = \sum_{e=m(\mathbf{x})}^{p} |\bar{c}^*_e|$.

The complexity analysis is based on ratio value $\frac{S_{m(\mathbf{x})}}{|LB_\mu|}$ computing, for the unfixed variables, the sum of their absolute reduced cost values over $|LB_\mu|$, where LB_μ is a lower bound on $\mu(\mathbf{x})$. Given a non-optimal phase solution $\mathbf{x}^h,\ h \geq 0$, a lower bound on the optimality parameter is $LB_\mu = -|\bar{c}^*_{m(\mathbf{x}^h)}|$. The next proposition determines how many phases are required to increase this lower bound by a factor greater than $1/2$. The same question is then answered for $1/n$ in the following proposition. In both cases, the number of phases required to reach their respective jumps is established as a function of ratio value $\frac{S_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|}$.

Proposition 12 (*[17, Lemma 12]*). *Let $h \geq 0$ denote a non-optimal phase number. A sequence of no more than $L(\mathbf{x}^h) := \min\left\{\left\lfloor \frac{2S_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|} \right\rfloor, n\right\}$ phases allows μ to jump by a factor over $1/2$.*

Proof. For non-optimal phase solution $\mathbf{x}^h$, we have $\mu^h \geq -|\bar{c}^*_{m(\mathbf{x}^h)}|$. Suppose some Type 2 cycle W^l at the end of phase $l \leq L(\mathbf{x}^h)$ has cardinality $|W^l| > \frac{2S_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|}$. Then,

$$\mu^{L(\mathbf{x}^h)} \geq \mu^l \geq \frac{-S_{m(\mathbf{x}^h)}}{|W^l|} > \frac{-S_{m(\mathbf{x}^h)}}{\frac{2S_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|}} = \frac{-|\bar{c}^*_{m(\mathbf{x}^h)}|}{2}.$$

Otherwise, all phase cycles in these $L(\mathbf{x}^h)$ phases have cardinalities at most $L(\mathbf{x}^h)$ arcs. Therefore, by (16),

$$\mu^{L(\mathbf{x}^h)} \geq \mu^h \left(1 - \frac{1}{L(\mathbf{x}^h)}\right)^{L(\mathbf{x}^h)} \geq -|\bar{c}^*_{m(\mathbf{x}^h)}| \left(1 - \frac{1}{L(\mathbf{x}^h)}\right)^{L(\mathbf{x}^h)} > \frac{-|\bar{c}^*_{m(\mathbf{x}^h)}|}{2}. \quad \square$$

Proposition 13 (*[17, Lemma 13]*). *Let $h \geq 0$ denote a non-optimal phase number. A sequence of no more than $O\left(\frac{nS_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|}\right)$ phases allows μ to jump by a factor over $1/n$.*

Proof. For non-optimal phase solution $\mathbf{x}^h$, we have $\mu^h \geq -|\bar{c}^*_{m(\mathbf{x}^h)}|$. As long as variable $x_{m(\mathbf{x}^h)}$ is not fixed, $S_{m(\mathbf{x}^h)}$ remains the same. By Proposition 12, we have $\mu > \frac{-|\bar{c}^*_{m(\mathbf{x}^h)}|}{2}$ after $\left\lfloor \frac{2S_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|} \right\rfloor$ phases, $\mu > \frac{-|\bar{c}^*_{m(\mathbf{x}^h)}|}{4}$ after $\left\lfloor \frac{2S_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|/2} \right\rfloor$ new phases,

Please cite this article in press as: J.B. Gauthier, et al., About the minimum mean cycle-canceling algorithm, Discrete Applied Mathematics (2014), http://dx.doi.org/10.1016/j.dam.2014.07.005

$\mu > \frac{-|\bar{c}^*_{m(\mathbf{x}^h)}|}{8}$ after $\left\lfloor \frac{2S_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|/4} \right\rfloor$ additional phases, and so on for the following steps increasing each time the lower bound by a factor greater than one half. The number of steps t such that $2^t \geq n$ is $t = \lceil \log n \rceil$, so that the total number of phases during these t steps is given by

$$\sum_{t=0}^{\lceil \log n \rceil - 1} \left\lfloor \frac{2S_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|/2^t} \right\rfloor \leq \lceil \log n \rceil + \sum_{t=0}^{\lceil \log n \rceil - 1} \frac{2S_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|/2^t} \leq \lceil \log n \rceil + \frac{2S_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|} \sum_{t=0}^{\lceil \log n \rceil - 1} 2^t$$
$$= \lceil \log n \rceil + \frac{2S_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|} \left(2^{\lceil \log n \rceil} - 1\right) \leq \lceil \log n \rceil + \frac{2S_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|}(2n - 1). \quad (17)$$

Hence, the number of phases needed to increase μ^h by a factor of over $1/n$ is $O\left(\frac{nS_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|}\right)$. □

The following theorem brings all these elements together. The idea is that it does not require the same amount of phases for each variable to be implicitly fixed. For some variables, the jump provided by Proposition 12 is sufficient while others must wait for that of Proposition 13.

Theorem 3 *([17, Theorem 1]).* *Given a capacitated network with arbitrary real-valued arc costs, MMCC performs $O(mn)$ phases.*

Proof. Starting with x_1 followed by $x_2, \ldots, x_p$, these variables are (implicitly) fixed one by one as μ increases. The fixing of x_e, $1 \leq e \leq p$, the yet unfixed variable with the largest absolute reduced cost value, to its lower or upper bound is done with Proposition 11. Observe that until x_{e+1} is fixed, $S_e = |\bar{c}^*_p| + \cdots + |\bar{c}^*_{e+2}| + |\bar{c}^*_{e+1}| + |\bar{c}^*_e|$ does not change.

The fixing of x_e is *fast* if $\frac{|\bar{c}^*_{e-1}|}{2} \leq |\bar{c}^*_e| \leq |\bar{c}^*_{e-1}|$, that is, the successive absolute reduced cost values are relatively close to each other. Otherwise, the fixing of x_e is *slow*.

By Proposition 12, for every variable x_e for which the fixing is fast, the number of phases to do so is at most n. Thus, the total number of phases of fast fixing for at most $p \leq m$ variables is $O(pn) \equiv O(mn)$.

Regarding the slow fixing process of a variable x_{e+1}, we have $|\bar{c}^*_{e+1}| < \frac{|\bar{c}^*_e|}{2}$, hence $\frac{|\bar{c}^*_{e+1}|}{|\bar{c}^*_e|} < \frac{1}{2}$. If the fixing process of x_{e+2} is also slow, we have $\frac{|\bar{c}^*_{e+2}|}{|\bar{c}^*_{e+1}|} < \frac{1}{2}$, hence $\frac{|\bar{c}^*_{e+2}|}{|\bar{c}^*_e|} = \frac{|\bar{c}^*_{e+2}|}{|\bar{c}^*_{e+1}|} \times \frac{|\bar{c}^*_{e+1}|}{|\bar{c}^*_e|} < \frac{1}{2^2}$. In the worst case, the fixing process is slow for at most p variables. By Proposition 13, the number of phases is bounded above by

$$\sum_{e=1}^{p} O\left(\frac{nS_e}{|\bar{c}^*_e|}\right) = O\left(n \sum_{e=1}^{p} \frac{S_e}{|\bar{c}^*_e|}\right)$$
$$= O\left(n \sum_{e=1}^{p} \frac{|\bar{c}^*_e| + |\bar{c}^*_{e+1}| + |\bar{c}^*_{e+2}| + \cdots + |\bar{c}^*_p|}{|\bar{c}^*_e|}\right)$$
$$= O\left(n \sum_{e=1}^{p} \left(1 + \frac{|\bar{c}^*_{e+1}|}{|\bar{c}^*_e|} + \frac{|\bar{c}^*_{e+2}|}{|\bar{c}^*_e|} + \cdots + \frac{|\bar{c}^*_p|}{|\bar{c}^*_e|}\right)\right)$$
$$< O\left(n \sum_{e=1}^{p} \left(1 + \frac{1}{2} + \frac{1}{2^2} + \cdots + \frac{1}{2^{p-e}}\right)\right)$$
$$< O\left(n \sum_{e=1}^{p} (1 + 1)\right) = O(pn) \equiv O(mn). \quad (18)$$

Whether it is fast or slow, the total implicit fixing process takes $O(mn)$ phases. □

Remark. The beauty of the slow analysis is that while the current candidate variable, x_e, may take longer to fix, it also implies that the optimal reduced cost distribution of the remaining non-fixed variables is skewed towards zero exponentially faster than the current candidate's optimal reduced cost. In other words, the higher number of phases required to implicitly fix x_e is eventually amortized by the much faster implicit fixing of the remaining $x_{e+1}, \ldots, x_p$ variables.

Radzik and Goldberg [17] also show this bound is tight by using certain minimum cost flow examples that behave as bad as the worst case complexity would have it. This means that the absolute reduced cost spread is such that arcs are implicitly fixed precisely at the bounds computed previously.

Please cite this article in press as: J.B. Gauthier, et al., About the minimum mean cycle-canceling algorithm, Discrete Applied Mathematics (2014), http://dx.doi.org/10.1016/j.dam.2014.07.005

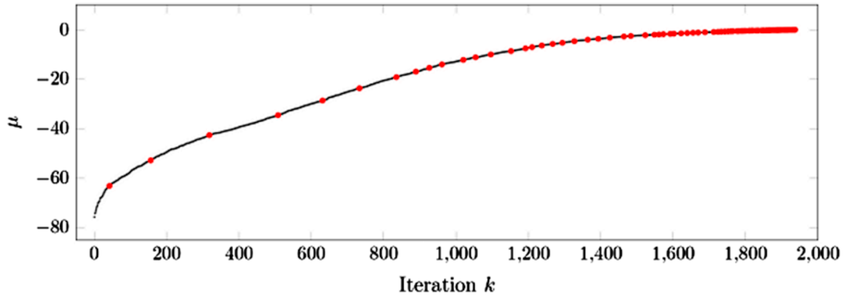

Fig. 5. Optimality parameter μ for Instance 1 [iteration base].

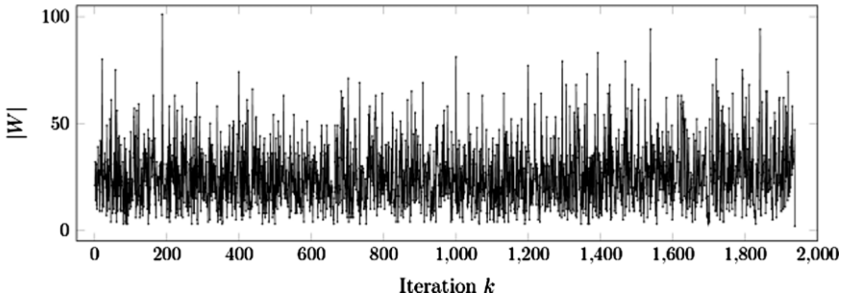

Fig. 6. Cycle size $|W|$ for Instance 1 [iteration base].

3.5. Bottleneck management

Although we have presented the complexity analysis in terms of phases, each of these can be seen as a group of iterations. As such, this section is separated in two parts. The first is reserved to the actual resolution of the pricing step while the second considers the phase as a whole, that is, using the Cancel-and-Tighten strategy. Several plots are presented to help grasp some of the ideas as well as appreciate the empirical behavior on a relatively large capacitated network flow problem comprising 1025 nodes and 91,220 arcs. We refer to the latter as Instance 1.

3.5.1. Iteration base

Solving the pricing step at every iteration corresponds to the traditional form of MMCC. Fig. 5 depicts the behavior of $\mu^k,\ k \geq 0$, on Instance 1: 1937 iterations are performed to reach optimality at $\mu = 0$. As expected, values of μ are non-decreasing satisfying Proposition 4. In Fig. 6, we see that the optimal cycle size $|W|$ however exhibits no pattern, a behavior quite different from the one observed for the maximum flow problem in Fig. 2.

For the purpose of the complexity analysis, the dynamic programming approach devised by Karp [15] runs in $O(mn)$ time. The proof of complexity is actually more simple than its design. While it is true that the first layer of the algorithm is based on dynamic programming, finding the optimal solution μ requires an additional layer of computations which break an important feature of the strategy matrix, that is, the knowledge of optimal strategies for each action state. If that is not enough, extracting the associated optimal cycle requires additional computations.

We also state without proof that it is a best-case complexity making the resolution of the pricing step systematically expensive. From a practical point of view, a word of caution is therefore in order. In fact, this problem is at the heart of many industrial challenges, see [6,5], or [12] for in-depth experimental analysis of different algorithms. Ahuja et al. [1, Chapter 5] also review several of them. While practice shows that the two algorithms of Howard [14] (as specialized by Cochet-Terrasson et al. [3]) and Young et al. [22] are top performers, their theoretical complexities are higher than that of Karp's.

Theorem 4 *([13, Theorem 3.10] and Radzik and Goldberg [17])*. *MMCC runs in $O(m^2n^2 \log n \log(nC))$ time for integer arc costs and $O(m^3n^2)$ time for arbitrary real-valued arc costs.*

Proof. The proof is immediate from the runtime complexity $O(mn)$ of each iteration combined with either Theorem 1 or Theorem 3 depending on the data type. Note that the theorems must first be translated back to iteration results. □

Regardless of the algorithm selected to solve the pricing step, it is still no match to the better design of Cancel-and-Tighten. In order to support this claim, we argue that the performance of the iteration based algorithm is vulnerable to the starting solution. We have launched the resolution process using $\mathbf{x}^0$ as the optimal solution of the maximization problem. The reader is now invited to consider the markings on Fig. 5. These markings indicate the starting point of each phase. Both cases traverse roughly the same number of phases, namely 90 and 87. The second launch actually requires 31,231 iterations

Please cite this article in press as: J.B. Gauthier, et al., About the minimum mean cycle-canceling algorithm, Discrete Applied Mathematics (2014), http://dx.doi.org/10.1016/j.dam.2014.07.005

to reach optimality at $\mu = 0$, yet the first 4 phases contain over 90% of the total iterations. Let us move on to a more integrated approach.

3.5.2. Phase base

We have already underlined the importance of the node potentials $\boldsymbol{\pi}^h$ established at the beginning of phase h for their ability to determine several Type 1 cycles. To appreciate the validity of alternative strategies, we underline that the complexity analysis addresses only the behavior of the optimality parameter; the actual path of resolution along with the corresponding primal solutions are irrelevant. There is in fact absolutely no reason for any two implementations to reach the same phase solution. From this rationale emanates what is fundamentally important in the complexity analysis: the phases. The importance of Proposition 4, the non-decreasing optimality parameter μ, is discarded along with the order of cancellation in favor of the strict increase from one phase to the next. This means that any strategy falling under the premises of a phase can benefit from the outer loop complexity analysis. The remainder of this section is dedicated to one such approach, namely *Cancel-and-Tighten* [13, Section 4]. Of course, the latter is subtle by nature as it aims to analyze the complexity of the whole phase using an efficient system carefully designed to ensure a phase is indeed delivered.

The name of Cancel-and-Tighten is self-explanatory of its two main steps: the cycle cancellations and the adjustment of node potentials. This strategy shines by the way these two steps are carried out. Recall that a phase is a group of sequential iterations defined by consecutive Type 1 cycles and terminated by a Type 2 cycle. The idea of the first step defers to the first part whereby only cycles of Type 1 are canceled. While the cliffhanger is not intentional, the repercussion of the Type 2 cycle is handled in the second step which refines the node potentials using the measurable jump property. Since the Cancel step focuses on Type 1 cycles, the idea of working on a subgraph that does the same is quite natural. Before moving to the technical aspects of the steps, let us describe the so-called *admissible* network.

Admissible network. Given $\boldsymbol{\pi}$, the nature of Type 1 cycles is to contain only negative reduced cost arcs with respect to these node potentials. Let us define the admissible network with respect to solution $\mathbf{x}$ accordingly: $G(\mathbf{x}, \boldsymbol{\pi}) := (N, A(\mathbf{x}, \boldsymbol{\pi}))$, where $A(\mathbf{x}, \boldsymbol{\pi}) := \{(i, j) \in A(\mathbf{x}) \mid \bar{d}_{ij} < 0\}$, that is, a residual arc is *admissible* if its reduced cost is strictly negative.

Cancel step. Let $h \geq 0$ denote a non-optimal phase number, $\mathbf{x}^h$ the solution at the beginning of the phase, and $\boldsymbol{\pi}$ some dual vector. By definition of the admissible network $G(\mathbf{x}^h, \boldsymbol{\pi})$, any and all cycles it contains are of Type 1. This means that by sequentially eliminating at most m Type 1 cycles, however arbitrary the order, one reaches some solution $\mathbf{x}$ which can be substantially different than the input $\mathbf{x}^h$. As the Cancel step progresses, the content of the admissible network becomes difficult to describe in mathematical terms because the notation looses track of the current solution. Nevertheless, recall that the reduced cost of every arc stays the same during the whole phase regardless of the cycles canceled because the node potentials are fixed to $\boldsymbol{\pi}$ throughout the step. In other words, as Type 1 cycles are canceled, only the residual capacities are modified and the admissible network is updated accordingly. The update is actually simpler to carry out than in the residual network. Indeed, by definition of admissibility, an arc and its reverse cannot be admissible simultaneously. This also means that the admissible network $G(\mathbf{x}, \boldsymbol{\pi})$ gets sparser because at least one new arc is saturated each time a Type 1 cycle is canceled.

Regardless of how the Cancel step is performed, one eventually reaches a solution $\mathbf{x}^{h+1}$ such that $G(\mathbf{x}^{h+1}, \boldsymbol{\pi})$ is acyclic. Of course, we have yet to prove optimality. In fact, we have yet to actually terminate the phase since the Type 2 cycle is still unconsidered. Let us see how this last operation is handled in the Tighten step.

Tighten step. Assume optimal values $[\boldsymbol{\pi}^h, \mu^h]$ are known and that the admissible network $G(\mathbf{x}^{h+1}, \boldsymbol{\pi}^h)$ is acyclic. By definition of a phase, we know the would be following iteration, say l, induces a Type 2 cycle W^l in the residual network $G(\mathbf{x}^{h+1})$. By Proposition 3, we also know that there exists some optimal vector of node potentials $\boldsymbol{\pi}^l$ such that all arcs on this cycle have the same negative reduced cost evaluated at μ^l. Observe that W^l is a Type 2 cycle with respect to selected $\boldsymbol{\pi} = \boldsymbol{\pi}^h$ but, by considering a new phase and modifying the node potentials to $\boldsymbol{\pi}^l$, this same cycle is a Type 1 cycle in $G(\mathbf{x}^{h+1}, \boldsymbol{\pi}^l)$. Let us rephrase this. The admissible network $G(\mathbf{x}^{h+1}, \boldsymbol{\pi})$ is defined by the current solution $\mathbf{x}^{h+1}$ and the reduced costs which are themselves defined by the selected vector $\boldsymbol{\pi}$ of node potentials. A solution $\mathbf{x}^{h+1}$ can induce different admissible networks depending on the selection of $\boldsymbol{\pi}$. The latter therefore sit at the top of the chain of command.

The Tighten step can be seen as the last operation that must be completed to start a new phase. Solving the pricing step effectively fetches the wanted information, that is, the best μ^{h+1} along with an optimal dual vector $\boldsymbol{\pi}^{h+1}$ which can be used for the duration of the next phase. This can of course be done with the dynamic programming approach of Karp [15]. Fig. 7 shows the result of the experiment we have carried on Instance 1. The resolution process requires 88 phases. Even though the node potentials are updated under one hundred times, the truth of the matter is that this approach is still too expensive. Nevertheless, the reader should think of this result as a reference convergence performance for Instance 1.

The alternative is to estimate both the new node potentials and the optimality parameter. However, in order for Cancel-and-Tighten to benefit from the complexity analysis, one must have a valid combination of such estimates at the beginning of each phase. Goldberg and Tarjan [13] call this the explicit maintenance of a price function $[\hat{\boldsymbol{\pi}}^h, \hat{\mu}^h]$, $h \geq 0$. Recall the dual version of the pricing problem (6)–(7). Once the node potentials are fixed, there is actually little room for $\hat{\mu}^h$. Indeed, $\hat{\mu}^h := \min_{(i,j)\in A(\mathbf{x}^h)} \hat{d}^h_{ij}$, where $\hat{d}^h_{ij} := d_{ij} - \hat{\pi}^h_i + \hat{\pi}^h_j$, which means that $\mu^h \geq \hat{\mu}^h$.

Given that μ^{h+1} would be the optimal solution to the pricing step at the current state of the algorithm (canceling a Type 2 cycle W^l), the goal is simple: establish $[\hat{\boldsymbol{\pi}}^{h+1}, \hat{\mu}^{h+1}]$ such that $\hat{\mu}^{h+1} = \mu^{h+1}$. Of course the latter is only wishful thinking

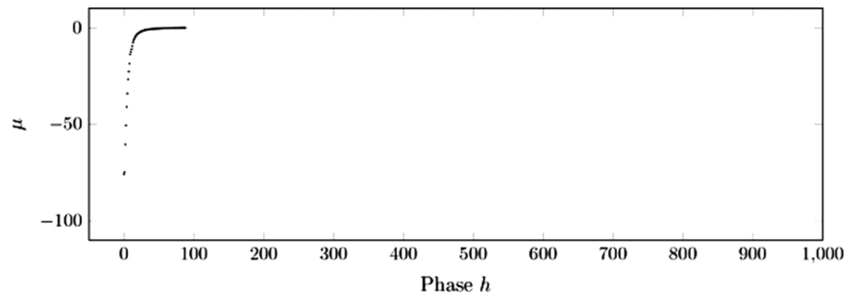

Fig. 7. Optimality parameter μ for Instance 1 [phase base—optimal].

but fortunately we still have one card up our sleeve. Assume $\underline{\mu}^{h+1}$ is a valid lower bound for μ^{h+1}. The new node potentials $\hat{\boldsymbol{\pi}}^{h+1}$ must therefore be such that

$$\hat{\mu}^{h+1} \geq \underline{\mu}^{h+1} \geq (1 - {}^{1}\!/_{n})\, \hat{\mu}^{h}. \tag{19}$$

The right-hand side inequality ensures that $\hat{\mu}^{h+1} > \hat{\mu}^{h}$ within the minimalist specifications of the $(1 - {}^{1}\!/_{n})$ jump property. The importance of the lower bound lies in its ability to transfer additional information to the node potentials in a constructive manner. As such, this lower bound not only aims to lift the optimality parameter as much as possible, it must also sport some intrinsic value with respect to the definition of a measurable jump obtained at the end of a phase. We provide a new approximation structure influenced by the tighter jump factor seen in (16).

In the spirit of the leading premise of the Tighten step, assume estimates $[\hat{\boldsymbol{\pi}}^{h}, \hat{\mu}^{h}]$ are readily available. Let us establish what we do know about the next minimum mean cycle. Since $G(\mathbf{x}^{h+1}, \hat{\boldsymbol{\pi}}^{h})$ is acyclic, it is possible to associate a level, L_i^h, $i \in N$, to each node using a topological ordering. These levels are recursively defined as

$$L_j^h := \max_{(i,j)\in A(\mathbf{x}^{h+1},\hat{\boldsymbol{\pi}}^{h})} L_i^h + 1,$$

with $L_i^h := 0$ if node i has no incoming admissible arcs. By construction of the ordering, if an arc $(i, j) \in A(\mathbf{x}^{h+1})$ is still admissible with respect to $\hat{\boldsymbol{\pi}}^{h}$ then the levels L_i^h and L_j^h are such that $L_j^h > L_i^h$. All other arcs have $L_j^h \leq L_i^h$.

Proposition 14. *The marginal update* $\hat{\pi}_i^{h+1} := \hat{\pi}_i^{h} - \frac{L_i^h}{L^h+1}\hat{\mu}^{h},\ \forall i \in N$, *yields a valid estimation for* $\hat{\mu}^{h+1}$. *Three possible values for* L^h *are*

$$L_{(a)}^h := n - 1, \qquad L_{(b)}^h := \max_{i\in N} L_i^h, \qquad L_{(c)}^h := \max_{(i,j)\in A(\mathbf{x}^{h+1})\mid L_i^h > L_j^h} L_i^h - L_j^h.$$

Proof. Since the levels start at 0, the value L^h can be seen as the length of a path in the admissible network. Update $L_{(a)}^h$ is the first of two implementations proposed by Goldberg and Tarjan [13] and obviously extracts nothing from the level values. Update $L_{(b)}^h$ can be seen as the length of the longest path in terms of the number of arcs. Update $L_{(c)}^h$ refines this value by checking in the residual network for the existence of an arc capable of inducing a cycle on the longest path. In order to show that all options for L^h induce valid lower bounds for μ^{h+1}, it suffices to realize that any arc added to create a cycle on the path referred by L^h has a non-negative reduced cost. Observe that these arc length values can be ordered as $L_{(a)}^h \geq L_{(b)}^h \geq L_{(c)}^h$.

The proof that the transformation is valid is reminiscent of (19) and shows that the inequality is indeed verified. The modified reduced costs are evaluated by

$$\hat{d}_{ij}^{h+1} := d_{ij} - \left(\hat{\pi}_i^h - \frac{L_i^h}{L^h+1}\,\hat{\mu}^h\right) + \left(\hat{\pi}_j^h - \frac{L_j^h}{L^h+1}\,\hat{\mu}^h\right) = \bar{d}_{ij}^h - \frac{L_j^h - L_i^h}{L^h+1}\,\hat{\mu}^h, \quad \forall (i,j) \in A(\mathbf{x}^{h+1}). \tag{20}$$

In the case of admissible arcs, we have $1 \leq L_j^h - L_i^h \leq L$. Therefore,

$$\hat{d}_{ij}^{h+1} \geq \hat{\mu}^h - \frac{L_j^h - L_i^h}{L^h+1}\,\hat{\mu}^h = \left(1 - \frac{L_j^h - L_i^h}{L^h+1}\right)\hat{\mu}^h \geq \left(1 - \frac{1}{L^h+1}\right)\hat{\mu}^h.$$

In the case of non-admissible arcs, we have $0 \leq L_i^h - L_j^h \leq L^h$. We also know that these arcs have a non-negative reduced cost. Therefore,

$$\hat{d}_{ij}^{h+1} \geq 0 - \frac{L_j^h - L_i^h}{L^h+1}\,\hat{\mu}^h = \frac{L_i^h - L_j^h}{L^h+1}\,\hat{\mu}^h \geq \left(1 - \frac{1}{L^h+1}\right)\hat{\mu}^h. \quad \square$$

Please cite this article in press as: J.B. Gauthier, et al., About the minimum mean cycle-canceling algorithm, Discrete Applied Mathematics (2014), http://dx.doi.org/10.1016/j.dam.2014.07.005

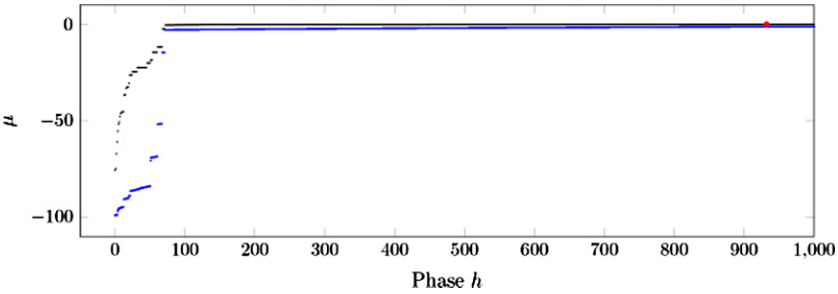

Fig. 8. Optimality parameter μ and estimate $\hat{\mu}$ for Instance 1 [phase base—update $L^h_{(a)}$].

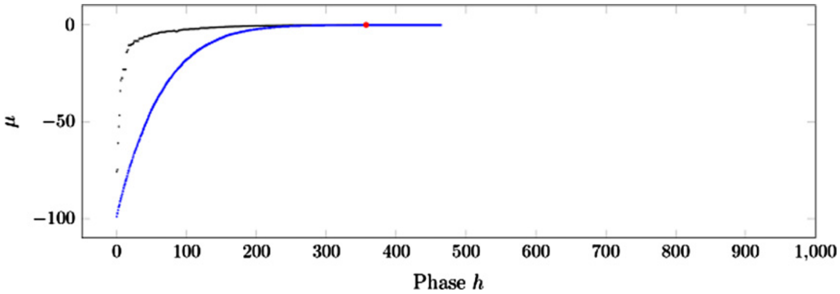

Fig. 9. Optimality parameter μ and estimate $\hat{\mu}$ for Instance 1 [phase base—update (d)].

Although we did not include the second implementation proposed in [13], call it Update (d), we have carried experiments using all four. The reasons for this omission are twofold. First, it does not perform as well as our two new update proposals $L^h_{(b)}$ and $L^h_{(c)}$. Second, it does not land itself naturally to our presentation with value L^h which is needed to reduce the complexity. The update is as follows: $\hat{\pi}^{h+1}_i := \hat{\pi}^h_i - qL^h_i,\ \forall i \in N$, where $q := \min_{(i,j)\in A(\mathbf{x}^{h+1})|L^h_i > L^h_j} \frac{\bar{d}^h_{ij} - \hat{\mu}^h}{L^h_i - L^h_j + 1}$. Observe that using value 0 instead of $\bar{d}^h_{ij}$ delivers Update $L^h_{(c)}$.

Initialization. Notice that while optimal values $[\boldsymbol{\pi}^h, \mu^h]$ do exist, the point of the approximation scheme is to carry out the computations with the approximation values $[\hat{\boldsymbol{\pi}}^h, \hat{\mu}^h]$ instead. As such, the explicit price function can be trivially initialized to $\hat{\pi}^0_i := 0,\ \forall i \in N$, and $\hat{\mu}^0 := \min_{(i,j)\in A(\mathbf{x}^0)} d_{ij}$.

The importance of properly updating the node potentials is capital to the speed of convergence. Indeed, the systematic computations over approximation values means that the associated errors are cumulative. Intuitively, a poor update of the node potentials induces a new admissible network that is not sufficiently altered to reveal new Type 1 cycles. Figs. 8–10 illustrate the point. Each figure contains three elements: two plots and a marking. The top level plot is the optimal value of μ^h retrieved for the sake of evaluating the quality of the estimate $\hat{\mu}^h$ as seen in the lower level plot. The marking indicates the phase number at which the optimal solution is reached although without proving it. Fig. 8 uses Update $L^h_{(a)}$ and suffers from an extremely poor convergence rate. There are in fact over 7000 additional phases required to prove optimality. In Fig. 9, we see that Update (d) performs much better than Update $L^h_{(a)}$. However, as can be seen in Fig. 10, it is still outperformed by Update $L^h_{(b)}$. These approximations require respectively 464 (343) phases to prove optimality. The results for Update $L^h_{(c)}$ are omitted because they are almost the same as those of Update $L^h_{(b)}$. As expected by the update mechanism, the estimate $\hat{\mu}^h$ is a lower bound for the optimal value μ^h. The quality of the update clearly influences how fast this bound is increased.

Proposition 15 (*[13]*). *The combination of the Cancel and Tighten steps runs in $O(m \log n)$ time.*

Proof. The Tighten step consists of a succession of basic operations on the arcs or the nodes, the most complex one being the topological ordering which runs in $O(m)$ time. The proof that the Cancel step terminates in $O(m \log n)$ time, making it the dominant method, is influenced by the works of Sleator and Tarjan [20]. The key lies in the realm of computer science whereby the sophisticated *splay tree* data structure allows for an efficient way to exhaustively search the admissible network. □

Using this strategy transcends the bottleneck incarnated by the pricing step with a convoluted approach. The bottleneck operation is now the completion of a phase which effectively allows the complexity analysis to trade the initial $O(m^2n)$ time per phase in favor of an amortized $O(m \log n)$ time. The improvement provided by the Cancel-and-Tighten strategy is the fruit of careful design. It is obtained by shedding another light on the iteration-wise analysis.

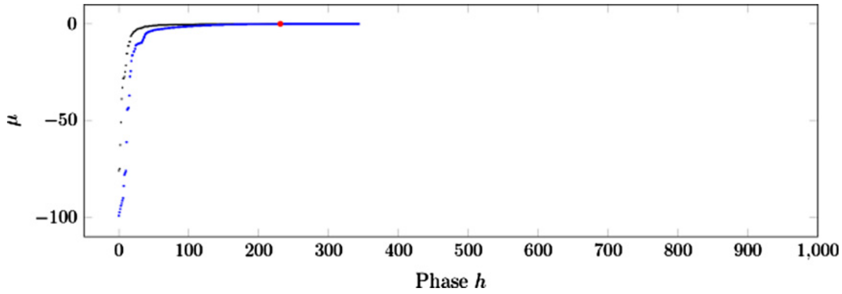

Fig. 10. Optimality parameter μ and estimate $\hat{\mu}$ for Instance 1 [phase base—update $L^h_{(b)}$].

While it is possible to fetch optimal node potentials at the end of every phase, this measure brings the global complexity to $O(n\log(nC))\times[O(m\log n)+O(mn)]\equiv O(mn^2\log(nC))$ for integer arc costs and $O(mn)\times[O(m\log n)+O(mn)]\equiv O(m^2n^2)$ for arbitrary real-valued arc costs.

The following two theorems present the global runtime complexity of MMCC when the Cancel-and-Tighten strategy is incorporated along with the approximation scheme. Although we have separated these theorems to highlight our new proposal in the second one, the reader is invited to read them as one. The idea behind their proofs is to place the resolution process in the same conditions as the complexity analysis of the outer loop. Since $\hat{\mu}^h \le \mu^h$, it is conceivable to rewrite Propositions 9 and 11 using approximation values $[\hat{\boldsymbol{\pi}}^h, \hat{\mu}^h]$ instead. Indeed, arc fixing then occurs in a more conservative fashion.

Theorem 5 *([13, Theorem 4.3]).* *MMCC accompanied by the Cancel-and-Tighten strategy runs in $O(mn\log n\log(nC))$ time for integer arc costs and $O(m^2n(\log n)^2)$ time for arbitrary real-valued arc costs.*

Proof. With respect to integer arc costs, the adaptation is straightforward. It turns out that the approximation values $\hat{\mu}$ follow the same geometric progression as the optimal ones. Indeed, the construction of the proof of Theorem 1 also holds with the approximation values. This means that $\hat{\mu} > -1/n$ is a valid stopping criterion and $O(n\log(nC))$ phases are performed.

In the case of arbitrary real-valued arc costs, we consider here the point of view of Theorem 2. It is therefore sufficient to have a measurable jump of $(1-1/n)$ at the end of each phase to obtain the wanted complexity. In the Tighten step, irrespectively of whether these phases are approximated or not, this same property is verified by construction, see (19). The approximation values therefore still follow the same behavior as the optimal ones. Hence, the number of phases using the Cancel-and-Tighten strategy is also $O(mn\log n)$. As far as the stopping criterion is concerned, polling for the optimal value of μ every n-th phase to assert the optimality certificate can effectively be done without compromising the complexity result. Indeed, the $O(mn)$ runtime of this operation can be discarded with respect to the amortization against the runtime of the $n-1$ previous approximated phases, i.e., $O((n-1)m\log n+mn)\equiv O(mn\log n)$.

Since each phase runs in $O(m\log n)$ time, the proof is brought to terms with its statement. □

Theorem 6. *MMCC accompanied by the Cancel-and-Tighten strategy runs in $O(m^2n\log n)$ time for arbitrary real-valued arc costs.*

Proof. From the point of view of Theorem 3, the use of the explicit price function $[\hat{\boldsymbol{\pi}}, \hat{\mu}]$ in accordance with Proposition 14 commands another look at Proposition 12. The following modified version of the proof basically suggests that a valid size indicator, say L^h+1, for the jump is sufficient, the actual size of the Type 2 cycle is not really needed.

Assume all update coefficients $L^h+1\le L(\mathbf{x}^h)$ in these $L(\mathbf{x}^h)$ phases. Then, by (16),

$$\hat{\mu}^{L(\mathbf{x}^h)} \ge -|\bar{c}^*_{m(\mathbf{x}^h)}|\left(1-\frac{1}{L(\mathbf{x}^h)}\right)^{L(\mathbf{x}^h)} > \frac{-|\bar{c}^*_{m(\mathbf{x}^h)}|}{2}.$$

Otherwise, some coefficient L^l+1 at the end of phase $l\le L(\mathbf{x}^h)$ has cardinality $L^l+1>\frac{2S_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|}$. Then,

$$\hat{\mu}^{L(\mathbf{x}^h)} \ge \hat{\mu}^l \ge \frac{L^l\,\mu^h}{L^l+1} \ge \frac{-S_{m(\mathbf{x}^h)}}{L^l+1} > \frac{-S_{m(\mathbf{x}^h)}}{\frac{2S_{m(\mathbf{x}^h)}}{|\bar{c}^*_{m(\mathbf{x}^h)}|}} = \frac{-|\bar{c}^*_{m(\mathbf{x}^h)}|}{2}.$$

Combine this with Proposition 13 and the $O(mn)$ result of Theorem 3 still stands. The global complexity ensues once the per phase runtime of $O(m\log n)$ is accounted for. The polling argument for the optimal value of μ is still applicable for the stopping criterion. □

Remark. While the original intent of Update (d) is not aligned with that of the approximation structure proposed, a value $L^h_{(d)}$ could still be extracted from the determination of the value q and therefore would still have the same complexity as the other updates listed.

Table 1
Complexity analysis summary.

Point of view	Outer loop		Global runtime complexity	
	No. of iterations	No. of phases	Cancel-and-Tighten strategy	
			Without	With
Theorem 1	$O(mn\log(nC))$	$O(n\log(nC))$	$O(m^2n^2\log(nC))$	$O(mn\log n\log(nC))$
Theorem 2	$O(m^2n\log n)$	$O(mn\log n)$	$O(m^3n^2\log n)$	$O(m^2n(\log n)^2)$
Theorem 3	$O(m^2n)$	$O(mn)$	$O(m^3n^2)$	$O(m^2n\log n)$

3.6. Summary and observations

The first part of this section assembles all the complexity results we have seen while the second raises several worthy observations about technical and mechanical aspects of the algorithm.

Table 1 summarizes the content of the complexity analysis. The first two columns display the complexity of the outer loop of MMCC depending on whether one thinks in terms of $O(m^2n)$ iterations or $O(mn)$ phases. An implementation which uses the Cancel-and-Tighten strategy grants a better theoretical complexity by consolidating several bottleneck operations. The last two columns show the global complexity of MMCC depending on whether or not it is incorporated.

Observe that the better complexity achieved by the integration of the Cancel-and-Tighten strategy within the MMCC framework is not a contradiction with the tight complexity of Radzik and Goldberg [17]. It is rather a testament to the importance of careful design. Indeed, the tight bound of $O(m^2n)$ iterations is superseded by the equivalent one of $O(mn)$ phases.

Aside from the theoretical improvements, the very essence of Cancel-and-Tighten exudes efficiency on several fronts. First, its conception allows for a more straightforward approach to the identification of negative cycles which further benefits from running on a sparser graph, that is, $A(\mathbf{x}^{h+1}, \hat{\boldsymbol{\pi}}^h) \subseteq A(\mathbf{x}^h, \hat{\boldsymbol{\pi}}^h) \subseteq A(\mathbf{x}^h)$. Second, the data structure allows better redundancy control than the iteration base approach which systematically restarts from scratch. Third, the ability to reuse information regarding the node potentials is not only appealing but also proves to be useful. Finally, its design matches the critical component of the complexity analysis and only aims to reach these important Type 2 cycle checkpoints as fast as possible.

Explicit arc fixing. The strongly polynomial complexity is obtained by introducing the concept of arc fixing. Truth be told, the bidirectional verification of the alternative statement for Proposition 8 can actually be carried out in any vanilla implementation of MMCC without compromising the resolution process nor the theoretical complexity. Fixed arcs can be removed from the system thus giving the rule a very practical effect. The arc fixing rule provided by Proposition 8 (or Proposition 9, extended to phases) can be, for all intent and purposes, *explicit*. Furthermore, observe that arc fixing is reserved for non-free variables. When $\mu = 0$, optimality is achieved. Post applying Proposition 8 would imply *all* variables get fixed regardless of their status. We underscore the fact that, in Proposition 10 (or Proposition 11, extended to phases), arc fixing is invariably implicit because it depends on an unknown optimal set of node potentials.

Cancellations. Recall that the iteration base necessitates 1937 iterations to terminate (Fig. 5). This means that the same number of cycles are canceled. Whether the phase base contains more or less cancellations is matter of resolution course but it is possible to give meaning to what we have observed. Let us speak numbers. In the optimal update method (Fig. 7), 4176 Type 1 cycles are canceled while in the three approximations (Figs. 8–10), these numbers are respectively 2192 and 5202 and 4417. The first approximation is strikingly different from the rest and actually is much closer to the iteration base number. Let us address this case first. With a poor update of the node potentials, the content of the admissible network is limited to a small fraction of arcs. This means that although we are not looking expressly for it, the minimum mean cycle is more likely to be identified. In other words, the Cancel-and-Tighten strategy behave similarly to the iteration base. In the other two approximations, the admissible network contains a lot of negative arcs and Type 1 cycles are identified in a random manner. It appears natural that more cycles are identified in this way than does the meticulous process of finding the best ones sequentially. What comes out of this interpretation is that it is more important to identify negative cycles fast than it is identifying the *best* one.

Tailing-off. The iteration base suffers from a tailing-off effect which can be explained by the nature of line search optimization and also that of MMCC. Regardless of the quality of the solution, all the little improvements must be accounted for before granting the optimality certificate. Since the optimality parameter is a gage for the expected improvement and that it converges to zero from below, the end of the resolution process is very much like a quest for crumbs. In the phase base, notice that all three approximation updates also suffer from a tailing-off effect which is even present in the optimal update method. Since the optimality parameter still intervenes, the same explanation holds as well for this approach. But there is more. The quality of the update plays a great role in shortening the tail. Indeed, the latter dictates both the content of the admissible network and the distance to optimality.

Switch offs. In practice, the tailing-off effect leads to believe that, when the optimality parameter reaches a very small value, it might be worthwhile to switch off to the iteration base. Indeed, as the process nears the optimal solution, the number of negative cycles becomes very small which makes the content of the admissible network even more limited in terms of

Type 1 cycles. Furthermore, the important graph reduction induced by the arc fixing dramatically reduces the computational penalty of the iteration base. On another note, the usefulness of Update $L^h_{(c)}$ is still unclear but the otherwise lack of tractable benefit suggests it might be wise to postpone its usage until the first variables get fixed in order to limit the impact of the additional $O(m)$ time it requires.

Dantzig–Wolfe decomposition. The decomposition of the residual problem (2)–(4) using a Dantzig–Wolfe decomposition scheme [4] brings yet another perspective to Cancel-and-Tighten. Let $k \geq 0$ denote a non-optimal iteration number. At every iteration, define the master problem as the set of upper bound constraints $y_{ij} \leq r^k_{ij},\ (i,j) \in A(\mathbf{x}^k)$, and formulate the subproblem for finding the minimum reduced cost. This results in the proposed pricing step defined by the primal–dual pair of linear programs (6)–(7) and (8)–(11). Its solution provides an improving cycle for which the maximum step size is computed using the master problem constraints. Therefore, Cancel-and-Tighten corresponds to heuristic solutions of that pricing problem, indeed a partial pricing devoted to Type 1 cycles only. An optimal solution to the residual problem (2)–(4) can only be guaranteed by solving the pricing subproblem to optimality and finding $\mu = 0$.

4. Conclusion

This paper aims to present the minimum mean cycle-canceling algorithm in its entirety by regrouping the knowledge from different sources. Ranging from the objective function to implicit arc fixing, a key component which traverses most of the proofs is a line of rationalization which questions the existence of admissible solutions. This admissibility defers to a fundamental piece of network theory, namely the flow decomposition principle. In the end, the same algorithm has been studied under numerous angles, each one providing theoretical breakthroughs. A very interesting point is that better bounds are obtained via very practical observations such as tighter jump factors or well conditioned admissible networks.

The original purpose of this work was a literature review preliminary to further researches, but we have come to see it as more than a summary. First of all, we harness the power of duality to simplify one of the building blocks of the algorithm. Secondly, the new way to look at the analysis directly in terms of $O(mn)$ phases is elegant in itself. We feel that we have integrated Cancel-and-Tighten to the minimum mean cycle-canceling framework under a new perspective and even contributed to its performance with new update approximations in the Tighten step. This in turn grants the reduction of the global runtime to $O(m^2 n \log n)$ and speaks volume about the importance of thinking in terms of phases. This third contribution is made possible by the generalization of Proposition 12. Finally, the computation results are enlightening of the resolution course of the minimum mean cycle-canceling algorithm. The different observations which come out of this study serve the practical side of things by reducing the wall-clock time of any generic implementation.

As a final note, this work is part of a much broader plan which includes generalizations to linear programming as well as understanding the ramifications with the *Improved Primal Simplex* method [8,18] in order to extract necessary adjustments required to recuperate some of the properties established herein.

Acknowledgment

Jacques Desrosiers acknowledges the Natural Sciences and Engineering Research Council of Canada for its financial support.

References

[1] Ravindra K. Ahuja, Thomas L. Magnanti, James B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, 1993.
[2] Dimitri Panteli Bertsekas, A distributed algorithm for the assignment problem, Technical Report, Laboratory for Information and Decision Systems Working Paper, MIT, Cambridge, MA, 1979.
[3] Jean Cochet-Terrasson, Guy Cohen, Stéphane Gaubert, Michael Mc Gettrick, Jean-Pierre Quadrat, Numerical computation of spectral elements in max-plus algebra, in: IFAC Conference on System Structure and Control, Nantes, France, July 8–10 1998.
[4] George B. Dantzig, Philip Wolfe, Decomposition principle for linear programs, Oper. Res. 8 (1) (1960) 101–111. http://dx.doi.org/10.1287/opre.8.1.101.
[5] Ali Dasdan, Experimental analysis of the fastest optimum cycle ratio and mean algorithms, ACM Trans. Des. Autom. Electron. Syst. 9 (4) (2004) 385–418. http://dx.doi.org/10.1145/1027084.1027085.
[6] Ali Dasdan, Sandy S. Irani, Rajesh K. Gupta, Efficient algorithms for optimum cycle mean and optimum cost to time ratio problems, in: 36th Design Automation Conference, 1999. Proceedings, 1999, pp. 37–42. http://dx.doi.org/10.1109/DAC.1999.781227.
[7] Jack Edmonds, Richard Manning Karp, Theoretical improvements in algorithmic efficiency for network flow problems, J. ACM 19 (2) (1972) 248–264. http://dx.doi.org/10.1145/321694.321699.
[8] Issmail Elhallaoui, Abdelmoutalib Metrane, Guy Desaulniers, François Soumis, An improved primal simplex algorithm for degenerate linear programs, INFORMS J. Comput. 23 (2011) 569–577. http://dx.doi.org/10.1287/ijoc.1100.0425.
[9] Lester Randolph Ford Jr., Delbert Ray Fulkerson, Maximal flow through a network, Canad. J. Math. 8 (1956) 399–404. http://dx.doi.org/10.4153/CJM-1956-045-5.
[10] Lester Randolph Ford Jr., Delbert Ray Fulkerson, Flows in Networks, Princeton University Press, Princeton, NJ, USA, 1962.
[11] Satoru Fujishige, A capacity-rounding algorithm for the minimum cost circulation problem: A dual framework of Tardos' algorithm, Math. Program. 35 (1986) 298–308. http://dx.doi.org/10.1007/BF01580882.
[12] Loukas Georgiadis, Andrew V. Goldberg, Robert Endre Tarjan, Renato F. Werneck, An experimental study of minimum mean cycle algorithms, in: Irene Finocchi, John Hershberger (Eds.), 2009 Proceedings of the Eleventh Workshop on Algorithm Engineering and Experiments (ALENEX), SIAM, 2009, pp. 1–13. http://dx.doi.org/10.1137/1.9781611972894.1.
[13] Andrew V. Goldberg, Robert Endre Tarjan, Finding minimum-cost circulations by canceling negative cycles, J. ACM 36 (4) (1989) 873–886. http://dx.doi.org/10.1145/76359.76368.
[14] Ronald Arthur Howard, Dynamic Programming and Markov Processes, The MIT Press, Cambridge, MA, USA, 1960.

[15] Richard Manning Karp, A characterization of the minimum cycle mean in a digraph, Discrete Math. 23 (3) (1978) 309–311. http://dx.doi.org/10.1016/0012-365X(78)90011-0.
[16] Morton Klein, A primal method for minimal cost flows with applications to the assignment and transportation problems, Manag. Sci. 14 (3) (1967) 205–220.
[17] Tomasz Radzik, Andrew V. Goldberg, Tight bounds on the number of minimum-mean cycle cancellations and related results, Algorithmica 11 (3) (1994) 226–242. http://dx.doi.org/10.1007/BF01240734.
[18] Vincent Raymond, François Soumis, Dominique Orban, A new version of the improved primal simplex for degenerate linear programs, Comput. Oper. Res. 37 (1) (2010) 91–98. http://dx.doi.org/10.1016/j.cor.2009.03.020.
[19] Hans Röck, Scaling techniques for minimal cost network flows, in: Uwe Pape (Ed.), Discrete Structures and Algorithms, Carl Hanser, Munich, 1980, pp. 181–191.
[20] Daniel Dominic Kaplan Sleator, Robert Endre Tarjan, A data structure for dynamic trees, J. Comput. System Sci. 26 (3) (1983) 362–391. http://dx.doi.org/10.1016/0022-0000(83)90006-5.
[21] Éva Tardos, A strongly polynomial minimum cost circulation algorithm, Combinatorica 5 (3) (1985) 247–255. http://dx.doi.org/10.1007/BF02579369.
[22] Neal E. Young, Robert Endre Tarjan, James B. Orlin, Faster parametric shortest path and minimum balance algorithms, Networks 21 (2) (1991) 205–221. http://dx.doi.org/10.1002/net.3230210206.

Please cite this article in press as: J.B. Gauthier, et al., About the minimum mean cycle-canceling algorithm, Discrete Applied Mathematics (2014), http://dx.doi.org/10.1016/j.dam.2014.07.005