INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2025, VOL. 71, NO. 4, PP. 1–9
Manuscript received July 31, 2025; revised October 2025. doi: 10.24425/ijet.2025.155486

Generalized Benders Decomposition to solve a nonlinear routing problem with queueing delay goal function

Kacper Kozerski, Andrzej Karbowski

***Abstract*—We address the multicommodity flow problem with a nonlinear goal function modeling queueing delay. It is well-known that linear programming solvers perform better than those used for nonlinear programming. We can leverage their performance by employing the Generalized Benders Decomposition (GBD) to partition the problem into master and primal subproblems. We prove that in the case of multiple subproblems, which is true in our case, we can split both the optimality and feasibility cuts and add them independently. Moreover, we extended a known proof of convergence to enable a wider range of problems to be solved using GBD. We use the split cuts technique to precompute feasibility cuts and analytically solve the subproblems to omit the use of nonlinear optimization software. Furthermore, we explore the possibilities of starting point selection through linear and quadratic approximation. We carry out tests on a classical network example to show that GBD can sometimes outperform nonlinear solvers, and also that quadratic approximation for starting point selection can provide strictly better solution times, dominating commercial solvers.**

***Keywords*—benders; GBD, optimization; decomposition; flow optimization; queueing delay**

I. Introduction

In the Internet age, we are faced with supplying a large number of users with data sent from different servers, with as little delay as possible. In order to achieve high-quality of service, routing algorithms have to account for interactions between different commodities flowing through the same links [1]. This aspect is also taken into account when planning and dimensioning network infrastructure [2].

Some authors consider game-theoretical formulations to reach a balance between different flows, for example, coalition games in international phone rerouting [3] or to determine behavior with overflowing packets [4]. To gain insight into the connection behavior under different loads, stemming from many flows using the same link, we can also employ queueing theory models [5] [6], which enable us to find the parameters such as sojourn time. Additionally, simulation and optimization are used to plan better network performance [7] [8]. In most cases, the models are linear, they often also have some discrete variables and we deal with mixed-integer linear programming (MILP) problems [9] [8]. This is partially motivated by the availability of computationally efficient optimization software for MILP problems like Xpress, Gurobi, or CPLEX. The linearity assumption can, however, be seen as limiting. For instance, queueing-theory-based models are often nonlinear [6].

Decomposition techniques can be employed to deal with computationally demanding problems. Augmented Lagrangian methods can deal with problematic constraints [10]. In the field of packet routing, it has found use in speeding up large-scale multicommodity routing [11]. Algorithms like column generation help us when too many variables are present [12], for instance, in the problem of routing and scheduling in multi-hop networks [13]. Benders Decomposition [14] and Generalized Benders Decomposition [15], [16] allow us to partition variable sets and create smaller subproblems. In the last decade, many papers on Benders Decomposition in routing have been published, as it allows, for instance, to solve efficiently energy-aware routing optimization [17].

This paper's goal is to bridge the gap between the important properties of nonlinear, queue-based models and fast, linear programming solvers. This is achieved through decomposing a multicommodity flow problem with Generalized Benders Decomposition. An important property of path-based flow problems is exploited, as the problem decomposes into many independent subproblems. We prove that in such a case, optimality and feasibility can be split and added independently of each other. Additionally, split cuts lead to a new criterion for algorithm convergence. We also propose methods to precompute the feasibility cuts in the case of a single-dimensional constraint function and solving the subproblems without the need to use nonlinear optimizers. We also explore different possibilities for obtaining a starting point selection for GBD.

II. Background

A. Network flow model

We take the modified formulation of multicommodity network flow, minimizing the queueing delay from [6]. It is a nonlinear programming problem. We use the notation, which can be found in Table I.

K. Kozerski and A. Karbowski are with Faculty of Electronics and Information Technology, Warsaw University of Technology, Warsaw, Poland (e-mail: kozerskikacper@gmail.com, andrzej.karbowski@pw.edu.pl).

$$\min_{x\in\mathbb{R}^{|W|},f\in\mathbb{R}^{|L|}} \quad \sum_{l\in L}\frac{f_l}{C_l-f_l+\varepsilon} \tag{1}$$

$$\sum_{w\in W}\sum_{p\in P_w}\mathbf{1}_p(l)\cdot x_p \le f_l \quad \forall l\in L \tag{2}$$

$$\sum_{p\in P_w} x_p = r_w \quad \forall w\in W \tag{3}$$

$$f_l \le C_l \quad \forall l\in L \tag{4}$$

$$x_p \ge 0 \quad \forall p\in P_w,\ w\in W \tag{5}$$

TABLE I
NOTATION USED IN THE MODEL

w	single flow (demand, connection) between a given source and a destination node
W	set of flows
P_w	set of all paths for the flow $w\in W$
L	set of all links of the network graph
C_l	flow rate capacity of the link l
r_w	demand of the flow rate for the flow $w\in W$
x_p	variable determining how much of demand w should be sent through path p (flow rate)
f_l	variable determining the flow rate in the arc $l\in L$
$\varepsilon>0$	small constant
$\mathbf{1}_p(l)$	a predicate, a function equal 1 when the link l belongs to path p and 0 otherwise.

The function (1) represents the sum of queueing delays on links. It grows monotonically with the flow rates in links f_l. The constraint (2) connects link flows f_l with routing decisions, the constraint (3) guarantees that the demand will be fulfilled and finally the constraint (4) guarantees no overflow on the link l.

We notice that if we fix the values of x variables to x^*, the problem decomposes into independent problems for each $l\in L$:

$$\min_{f_l\in\mathbb{R}} \quad \frac{f_l}{C_l-f_l+\varepsilon} \tag{6}$$

$$\sum_{w\in W}\sum_{p\in P_w}\mathbf{1}_p(l)\cdot x_p^* \le f_l \tag{7}$$

$$f_l \le C_l \tag{8}$$

This structure can then be exploited using the GBD algorithm.

B. Generalized Benders Decomposition theory

Generalized Benders Decomposition allows us to partition the variable sets into master and subproblem variables [15].

We first formulate a general optimization problem in the form [16]:

$$\min_{x\in X\subseteq\mathbb{R}^n, v\in V\subseteq\mathbb{R}^q} f(x,v) \tag{9}$$

s.t.

$$g(x,v)\le 0 \tag{10}$$

where:

$$f: \quad \mathbb{R}^n\times\mathbb{R}^q\to\mathbb{R} \tag{11}$$

$$g: \quad \mathbb{R}^n\times\mathbb{R}^q\to\mathbb{R}^m \tag{12}$$

We can reformulate this problem in the following manner:

$$\min_{v\in V\cap V_0} z(v) \tag{13}$$

$$V_0 \triangleq \{v: \quad \exists x\in X \quad g(x,v)\le 0\} \tag{14}$$

$$z(v)\triangleq \inf\Big\{f(x,v): x\in X, \quad g(x,v)\le 0\Big\} \tag{15}$$

The set V_0 guarantees feasibility of the primary subproblem (V_0 is referred to as a solvability set).

Theorem 1. *Projection [15]*

1) *Problem (9)-(10) is infeasible iff problem (13)-(14) is infeasible*
2) *Problem (9)-(10) is unbounded iff problem (13)-(14) is unbounded*
3) *If $(\hat{x},\hat{v})$ is the optimal solution of the problem (9)-(10), then $\hat{v}$ is the optimal solution of the problem (13)-(14)*
4) *If $\hat{v}$ is the optimal solution of the problem (13)-(14) and $\hat{x}$ reaches its infimum at $v=\hat{v}$, then $(\hat{x},\hat{v})$ is the optimal solution of the problem (9)-(10)*

Theorem 2. *Representation of V_0 [15]*
Let us assume that:

1) *X is a nonempty convex set*
2) *$g(x,v)$ is convex on $x\in X$ for each fixed $v\in V$*
3) *For each fixed $v\in V$, the set:*

$$Z_v=\{z\in\mathbb{R}^m: \exists x\in X \quad g(x,v)\le z\} \tag{16}$$

is closed

Then a point $v^*\in V$ belongs to the set V_0 iff:

$$\sup_{\lambda\in\Lambda_f}\inf_{x\in X} L_f(x,v^*,\lambda)\le 0 \tag{17}$$

$$\Lambda_f \triangleq \{\lambda\in\mathbb{R}^m: \lambda\ge 0, \sum_{j=1}^{m}\lambda_j=1\} \tag{18}$$

and

$$L_f(x,v,\lambda)\triangleq \lambda^\top g(x,v) \tag{19}$$

Theorem 3. *Representation of z(v) [15]*
Let us assume that:

1) *X is a nonempty convex set*
2) *$g(x,v), f(x,v)$ are convex on X for fixed value of $v=v^*, v\in V$*
3) *For each $v^*\in V$ at least one of the following conditions is true:*
   a) *$z(v^*)$ is finite and there exists a vector of optimal Lagrange multipliers for the problem (13)-(14)*
   b) *$z(v^*)$ is finite, $g(x,v^*)$ and $f(x,v^*)$ are continuous on X, X is closed and there exists $\varepsilon\ge 0$, such that the ε-optimal solution set of (13)-(14) is nonempty and bounded*
   c) *$z(v^*)=+\infty$*

Then

$$z(v)=\sup_{\lambda\in\Lambda_o}\inf_{x\in X} L_o(x,v,\lambda) \tag{20}$$

where

$$\Lambda_o \triangleq \{\lambda\in\mathbb{R}^m: \lambda\ge 0\} \tag{21}$$

and

$$L_o(x, v, \lambda) \triangleq f(x, v) + \lambda^\top g(x, v) \tag{22}$$
$$v \in V \cap V_0 \tag{23}$$

Using Thms 1 - 3 we can formulate the problem (9) - (10) as:

$$\min_{v \in V, \mu \in \mathbb{R}} \mu \tag{24}$$

s.t.

$$\inf_{x \in X} L_o(x, v, \lambda) \leq \mu \qquad \forall \lambda \in \Lambda_o \tag{25}$$
$$\inf_{x \in X} L_f(x, v, \lambda) \leq 0 \qquad \forall \lambda \in \Lambda_f \tag{26}$$

As the sets in conditions (25) - (26) are of cardinality up to $\mathfrak{c}$, we cannot use this formulation directly. Instead, we iteratively substitute sets, therefore approximating the original formulation. The finite set versions of (25) - (26) are referred to as cuts. In every iteration, if the primal problem is feasible, we save the dual λ values and add them to the set, as they are from the set Λ_o. If the problem turns out infeasible, we solve the feasibility problem [16]:

$$\min_{x \in X, \alpha \in \mathbb{R}} \alpha \tag{27}$$

s.t.

$$g_j(x, v) \leq \alpha \qquad \forall j \in \{1..m\} \tag{28}$$

which supplies us with dual values from Λ_f.

Definition 1. *P property [15] (infimal independence)*
We say that the Lagrangian function $L(x, v, \lambda)$ satisfies P property when we can find the value of $\inf_{x \in X} L(x, v, \lambda)$ independently of the value of v.

Some examples of problems that result in infimally independent Lagrangians are problems with separable goal and constraint functions and variable factor programming problems [15].

P property allows us to omit explicit infima calculations in cuts. The optimal values of the cut Lagrangians are calculated while solving feasible primal problems or feasibility problems, since the optimality Lagrangian is dual to the feasible primal problem and the feasibility Lagrangian is dual to the feasibility problem. Thus, we can store the values of x^k, λ^k from iteration k and plug them into our Lagrangian functions.
Finally, we arrive at an iterative Algorithm 1.

The constraints (32)-(33) added in each iteration are, respectively, optimality and feasibility cuts.

Theorem 4. *Convergence of the GBD algorithm [15], [16]*
GBD algorithm converges in a finite number of steps with a given tolerance $\beta > 0$ when either:

1) *V is a finite, discrete set and the assumptions of $z(v)$ and V_0 representation Thms 2,3 are satisfied (even with $\beta = 0$)*
2) *V is a nonempty and compact set, $V \subseteq V_0$, X is a nonempty compact convex set, functions f and g are convex on X for each fixed $v \in V$ and continuous on $X \times V$, the set of optimal Lagrange multiplier of the subproblem is nonempty for fixed $v \in V$ and constraints satisfy Slater's regularity condition: $\exists x \in X \quad \exists v \in V : g(x, v) < 0$*
3) *$V \not\subseteq V_0$, constraint function g is linearly separable: $g(x, v) = g_1(x) + g_2(v)$, set X is defined using linear constraints, rest of the conditions as in p. 2.*

Algorithm 1 Generalized Benders Decomposition

$UBD \leftarrow +\infty$
$LBD \leftarrow -\infty$
$v^1 \leftarrow select(v \in V)$
$iter \leftarrow 1$
$K_o, K_f \leftarrow \{\}, \{\}$
while $UBD - LBD \geq \beta$ **do**

$$\min\{f(x, v^{iter}) : x \in X, g(x, v^{iter}) \leq 0\} \tag{29}$$

  if Problem (29) is feasible **then**
    Save x^{iter} and λ^{iter}
    $K_o \leftarrow K_o \cup \{iter\}$
    $UBD \leftarrow \min(UBD, f(x^{iter}, v^{iter}))$
  else

$$\min\{\alpha : x \in X, g_j(x, v^{iter}) \leq \alpha \quad j \in \{1..m\}\} \tag{30}$$

    Save x^{iter} and λ^{iter} from problem (30)
    $K_f \leftarrow K_f \cup \{iter\}$
  end if
  $iter \leftarrow iter + 1$

$$LBD \leftarrow \min_{v \in V, \mu \in \mathbb{R}} \mu \tag{31}$$

  s.t.

$$L_o(x^k, v, \lambda^k) = f(x^k, v) + (\lambda^k)^\top g(x^k, v) \leq \mu \quad \forall k \in K_o \tag{32}$$
$$L_f(x^k, v, \lambda^k) = (\lambda^k)^\top g(x^k, v) \leq 0 \quad \forall k \in K_f \tag{33}$$

  Save v^{iter} from the problem (31) - (33)
end while

III. Multicut GBD

A. Separable problems

As we have mentioned before, some problems have a special structure that we can exploit.
The separable problems have the form:

$$\min_{v \in V, x_1 \in X_1, \ldots, x_p \in X_p} f_0(v) + \sum_{i=1}^{p} f_i(x_i, v) \tag{34}$$

s.t.

$$g_i(x_i, v) \leq 0 \qquad \forall i \in \{1..p\} \tag{35}$$

Setting the value of master variables $v = v^*$, the subproblem is composed of p independent subproblems, which can be solved in parallel:

$$\min_{x_i \in X_i} f_i(x_i, v^*) \tag{36}$$

s.t.

$$g_i(x_i, v) \leq 0 \tag{37}$$

B. Fully multicut Benders formulation

In the case of separable problems both the optimality and feasibility cuts can be split. This is a generalization of the so-called L-shaped method, the Benders' method applied to (linear) stochastic programming problems with a specific structure, where the objective cuts are often being "split" [18]. We will prove in Thm 6 that such a technique can also be applied in the case of GBD and feasibility and optimality cuts for each problem can be added independently to every subproblem. We will earlier show in Thm 5, that the feasibility cuts can be split. We also provide examples showing that split feasibility cuts can lead to tighter approximations of the feasibility set (Example 1) and that split objective cuts can lead to tighter approximations of the goal function (Example 2).

Theorem 5. *Feasibility multicut formulation*
For problems in the form (34)-(35), we can split feasibility cuts.

Proof. Since the subvectors x_i are independent of each other after setting the values of v, we have a solvability set in the form:

$$V_0 = \bigcap_{i=1}^{p} V_{0i} \tag{38}$$

$$V_{0i} \triangleq \{v : \exists x_i \in X_i \quad g_i(x_i, v) \leq 0\} \qquad \forall i \in \{1..p\} \tag{39}$$

$$v \in V_0 \iff v \in V_{01} \wedge v \in V_{02} \wedge .. \wedge v \in V_{0p} \tag{40}$$

We also have that the set Z_v is closed iff for all $i \in \{1..p\}$ the sets:

$$Z_v^i = \{z \in \mathbb{R}^{m_i} : \exists x_i \in X_i \quad g_i(x_i, v) \leq z\} \tag{41}$$

are closed.
Now we can apply the V_0 Representation Thm 2 to every set $V_{0i} \quad \forall i \in \{1..p\}$:

$$v \in V_{0i} \iff \sup_{\lambda_i \in \Lambda_i} \inf_{x_i \in X_i} L_{fi}(x_i, v, \lambda_i) \leq 0 \tag{42}$$

$$L_{fi}(x_i, v, \lambda_i) \triangleq \lambda_i^\top g_i(x_i, v) \tag{43}$$

$$\Lambda_i \triangleq \{\lambda_i \in \mathbb{R}^{m_i} : \lambda_i \geq 0, \sum_{j=1}^{m_i} \lambda_{ij} = 1\} \tag{44}$$

□

Practically, in order to gain $\lambda_i \in \Lambda_i$, for any infeasible subproblem i, we can solve feasibility subproblem i:

$$\min_{x_i \in X_i, \alpha_i \in \mathbb{R}} \alpha_i \tag{45}$$

s.t.

$$g_{ij}(x_i, v) \leq \alpha_i \qquad \forall j \in \{1..m_i\} \tag{46}$$

We conclude from Thm 5 that the split feasibility cuts can approximate the feasibility set at least as good as the formulation without splits. Moreover, the split cuts can provide a tighter approximation of the feasibility set in a smaller number of iterations, as we show in Example 1.

Example 1.

$$\min_{x \in [0,1]^2, v \in \{0,1\}^2} -v_1 - v_2 + x_1 + x_2 \tag{47}$$

s.t.

$$3v_1 + v_2 - x_1 - 1 \leq 0 \tag{48}$$

$$0.5 + v_2 - x_2 \leq 0 \tag{49}$$

After setting the values of v that cause infeasibility, the resulting feasibility subproblems: one with x_1, the second with x_2, can be solved independently.

If we choose v to be the complicating variables, then the primal problem will be feasible only when we set the values of complicating variables to $v_1 = 0, v_2 = 0$. Suppose we choose $v_1 = 1, v_2 = 1$ as the starting point (which makes the primal problem infeasible).

In the case without splitting feasibility cuts, we have a feasibility problem in the form:

$$\min_{x \in [0,1]^2, \alpha \in \mathbb{R}} \alpha \tag{50}$$

s.t.

$$3 - x_1 \leq \alpha \tag{51}$$

$$1.5 - x_2 \leq \alpha \tag{52}$$

The solution to such a problem will be at $\alpha = 2, x_1 = 1, x_2 \in [0, 1]$, making the constraint (51) the only active one with dual value of $\lambda_1 = 1$.

The resulting cut will be in the form:

$$\begin{aligned} &\lambda_1 \cdot (3v_1 + v_2 - x_1 - 1) + \lambda_2 \cdot (0.5 + v_2 - x_2) \leq 0 \\ \equiv \quad &1 \cdot (3v_1 + v_2 - 1 - 1) + 0 \cdot (0.5 + v_2 - x_2) \leq 0 \\ \equiv \quad &3v_1 + v_2 \leq 2 \end{aligned} \tag{53}$$

which cuts all infeasible solutions with $v_1 = 1$, but leaves the solution $v_1 = 0, v_2 = 1$, which is infeasible.

In the case of split cuts, we have two subproblems.
The first subproblem has the form:

$$\min_{x_1 \in [0,1], \alpha_1 \in \mathbb{R}} \alpha_1 \tag{54}$$

s.t.

$$3 - x_1 \leq \alpha_1 \tag{55}$$

with solution at the point $x_1 = 1, \alpha_1 = 2$. The value of the dual multiplier will be $\lambda_1 = 1$, because it is the only one existing in this problem (and they have to sum up to 1).
The second subproblem has the form:

$$\min_{x_2 \in [0,1], \alpha_2 \in \mathbb{R}} \alpha_2 \tag{56}$$

s.t.

$$1.5 - x_2 \leq \alpha_2 \tag{57}$$

with solution at the point $x_2 = 1, \alpha_2 = 0.5$. The value of the dual multiplier will be $\lambda_2 = 1$, because it is the only one existing in this problem (and they have to sum up to 1).

The first problem produces the cut:

$$\begin{aligned} & 3v_1 + v_2 - x_1 - 1 \leq 0 \\ \equiv \quad & 3v_1 + v_2 - 1 - 1 \leq 0 \\ \equiv \quad & 3v_1 + v_2 \leq 2 \end{aligned} \tag{58}$$

which cuts all the points with $v_1 = 1$ (since $v_2 \in \{0, 1\}$ we have $3 \cdot 1 + v_2 > 2$ and v_1 is a binary variable).

The second problem produces the cut:

$$\begin{aligned} & 0.5 + v_2 - x_2 \leq 0 \\ \equiv \quad & 0.5 + v_2 - 1 \leq 0 \\ \equiv \quad & v_2 \leq 0.5 \end{aligned} \tag{59}$$

which cuts all the points with $v_2 = 1$ (since v_2 is a binary variable). Both cuts produce a tighter approximation of the feasibility set (in this example the direct representation, which allows only $v_1 = 0, v_2 = 0$).

Theorem 6. *Objective multicut formulation*
If all assumptions of feasibility cut splitting Thm 5 hold, and the goal function is separable with respect to each decision variable subvector x_i, then we can also split the objective function cuts.

Proof. We have a problem in the form:

$$\min_{x \in X, v \in V} [f(x, v) = \sum_{i=1}^{p} f_i(x_i, v)] \tag{60}$$

s.t.

$$g_i(x_i, v) \leq 0 \qquad \forall i \in \{1..p\} \tag{61}$$

Since for the fixed values of v, the subproblem i becomes fully independent from other subproblems, we can solve every subproblem i as:

$$\inf_{x \in X_i} f_i(x_i, v) \tag{62}$$

s.t.

$$g_i(x_i, v) \leq 0 \tag{63}$$

Therefore, the vector of optimal Lagrange multipliers (for every feasible subproblem) can also be split into independent subvectors:

$$\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_p \end{bmatrix}, \quad \forall i \in \{1..p\} \quad \lambda_i \in \Lambda_i \tag{64}$$

where

$$\Lambda_i \triangleq \left\{ \lambda_i \in \mathbb{R}^{m_i} : \lambda_i \geq 0 \right\} \qquad \forall i \in \{1..p\} \tag{65}$$

Optimality Lagrangian takes the form:

$$L_o(x, v, \lambda) = \sum_{i=1}^{p} L_{oi}(x_i, v, \lambda_i) \tag{66}$$

where

$$L_{oi}(x_i, v, \lambda_i) \triangleq f_i(x_i, v) + \lambda_i^\top g_i(x_i, v) \quad \forall i \in \{1..p\} \tag{67}$$

Then, going back to $z(v)$ representation:

$$\begin{aligned} & \min_{v \in V \cap V_0} \sup_{\lambda \geq 0} \inf_{x \in X} \sum_{i=1}^{p} L_{oi}(x_i, v, \lambda_i) = \\ & \min_{v \in V \cap V_0} \sum_{i=1}^{p} \sup_{\lambda_i \geq 0} \inf_{x_i \in X_i} L_{oi}(x_i, v, \lambda_i) \end{aligned} \tag{68}$$

□

We conclude from Thm 6 that the split objective cuts will provide an approximation of the objective Lagrangian at least as good as the formulation without splits. Moreover, the approximation with split cuts can sometimes be tighter, as we show in Example 2.

Example 2.

$$\min_{v \geq 1, \mu \in \mathbb{R}^2} (\mu_1 + \mu_2) \tag{69}$$

s.t.

$$v \leq \mu_1 \qquad \textit{cut from 1st iteration} \tag{70}$$

$$-v \leq \mu_2 \qquad \textit{cut from 1st iteration} \tag{71}$$

$$-v \leq \mu_1 \qquad \textit{cut from 2nd iteration} \tag{72}$$

$$v \leq \mu_2 \qquad \textit{cut from 2nd iteration} \tag{73}$$

The solution to this problem is at the point $v = 1, \mu_1 = 1, \mu_2 = 1, \quad LBD = 2v = 2$.

A problem without splitting objective cuts:

$$\min_{v \geq 1, \mu \in \mathbb{R}} \mu \tag{74}$$

s.t.

$$v - v = 0 \leq \mu \qquad \textit{cut from 1st iteration} \tag{75}$$

$$-v + v = 0 \leq \mu \qquad \textit{cut from 2nd iteration} \tag{76}$$

The solution to this problem is at the point $v \geq 1, \mu = 0, \quad LBD = 0$.

The resulting algorithm will be in the form given in Algorithm 2.

Even if some subproblem turns out to be infeasible, we can both add feasibility cut for the infeasible subproblem and valid objective cuts for other feasible subproblems - since the infeasibility of the subproblem i_1 doesn't influence the fact that the feasible subproblem i_2 (obtained with the same value of v) creates valid values of λ_i. Adding optimality cuts, even for the infeasible v^* can approximate the behavior of the objective function in some proximity of v^*, which can in turn be feasible.

As we have mentioned, this algorithm can be used in the case of our routing problem.

Algorithm 2 Generalized Benders decomposition with split cuts

$UBD \leftarrow +\infty$
$LBD \leftarrow -\infty$
$v^1 \leftarrow select(v \in V)$
$iter \leftarrow 1$
$K_o, K_f \leftarrow \{\}, \{\}$
while $UBD - LBD \geq \beta$ **do**
  for $i = 1..p$ **do**

$$\min\{f_i(x_i, v^{iter}) : x_i \in X_i, g(x_i, v^{iter}) \leq 0\} \tag{77}$$

    if Problem (77) is feasible **then**
      Save x_i^{iter} and λ_i^{iter}
      $K_o \leftarrow K_o \cup \{(iter, i)\}$
    else

$$\min_{x_i, \alpha_i}\{\alpha_i : x_i \in X_i, g_{ij}(x_i, v^{iter}) \leq \alpha_i \quad j \in \{1..m_i\}\} \tag{78}$$

      Save x_i^{iter} and λ_i^{iter} from problem (78)
      $K_f \leftarrow K_f \cup \{(iter, i)\}$
    end if
  end for
  if $\forall i = 1..p$ problem i was feasible **then**
    $UBD \leftarrow \min(UBD, f_0(v) + \sum_{i=1}^{p} f_i(x_i^{iter}, v^{iter}))$
  end if
  $iter \leftarrow iter + 1$

$$LBD \leftarrow \min_{v \in V, \mu \in \mathbb{R}} f_0(v) + \sum_{i=1}^{p} \mu_i \tag{79}$$

  s.t.

$$L_{oi}(x_i^k, v, \lambda_i^k) = f_i(x_i^k, v) \tag{80}$$
$$+ (\lambda_i^k)^\top g_i(x_i^k, v) \leq \mu_i \quad \forall (k,i) \in K_o \tag{81}$$
$$L_{fi}(x_i^k, v, \lambda_i^k) = (\lambda_i^k)^\top g_i(x_i^k, v) \leq 0 \quad \forall (k,i) \in K_f \tag{82}$$

  Save v^{iter} from problem (79) - (82)
end while

C. New convergence criterion for specific cases

In the case of multicut formulation, we can relax the assumptions of Thm 4 p. 2, effectively obtaining a new convergence criterion. We will prove in Thm 7 that, instead of requiring that $V \subseteq V_0$, we can assume that:

$$V_i \subseteq V_0 \vee m_i = 1 \quad \forall i \in \{1..p\} \tag{83}$$
$$V_i = \{v : \exists x_i \in X_i \quad g_i(x_i, v) \leq 0\} \quad \forall i \in \{1..p\} \tag{84}$$

and m_i refers to the number of g_i constraints: $g_i : X_i \times V \to \mathbb{R}^{m_i}$.

This criterion can be used in our problem, as the dimensionality of the constraints (7) is 1 in every subproblem $l \in L$.

Theorem 7. *Let us suppose that $v \notin V_0$, but the conditions for feasibility multicut formulation hold and for all independent subproblems i, with respect to feasibility, we have that $V \subseteq V_{0i} \vee m_i = 1$. Rest of the conditions as in Thm 4 p.2. Then the problem will converge in a finite number of steps.*

Proof. We are only interested in problems where $V \not\subseteq V_{0i}$, since for the rest of them the feasibility is guaranteed.

From assumptions, we have that $m_i = 1$, so we are dealing with single-constraint subproblems. And we know that for such problems, the feasibility requires only a single cut (since $|\Lambda_i| = 1$). Therefore, we can spend up to p iterations adding cuts that provide feasibility (if needed). The rest follows from Thm 4 p.2. □

IV. Specific features of GBD implementation to the network routing problem concerning queuing delays

A. Analytical solution to subproblems

In our case, we can speed up subproblem computation by precomputing feasibility cuts, and once the feasibility is guaranteed, we can solve the subproblems in $O(m)$ time.

In the case of split cuts, the V_0 Representation Thm 2, applied to the set V_{0l}, $l \in L$ is in form:

$$\inf_{f_l \leq C_l} \lambda_l \Big(\sum_{w \in W} \sum_{p \in P_w} \mathbf{1}_p(l) \cdot x_p - f_l \Big) \leq 0 \qquad \lambda_l \in \{1\} \tag{85}$$
$$\equiv \quad \sum_{w \in W} \sum_{p \in P_w} \mathbf{1}_p(l) \cdot x_p - \sup_{f_l \leq C_l} f_l \leq 0 \tag{86}$$
$$\equiv \quad \sum_{w \in W} \sum_{p \in P_w} \mathbf{1}_p(l) \cdot x_p \leq C_l \tag{87}$$

If we add those cuts, before the problem starts, we will always have feasible subproblems.

Then we can notice that since the goal function is differentiable and strictly increasing, the optimal solution to the subproblem will be at the lowest possible point, that is, when the constraint (7) is active. Moreover, we can find the dual values from KKT conditions:

$$L_{ol}(f_l, x, \lambda_l) = \frac{f_l}{C_l - f_l + \varepsilon} +$$
$$\lambda_l \Big(\sum_{w \in W} \sum_{p \in P_w} \mathbf{1}_p(l) \cdot x_p - f_l \Big) \tag{88}$$
$$\frac{\partial L_{ol}}{\partial f_l}(f_l, x, \lambda_l) = 0 \tag{89}$$
$$\frac{C_l + \varepsilon}{(C_l - f_l + \varepsilon)^2} - \lambda_l = 0 \implies \lambda_l^* = \frac{C_l + \varepsilon}{(C_l - f_l^* + \varepsilon)^2} \tag{90}$$
$$f_l^* = \sum_{w \in W} \sum_{p \in P_w} \mathbf{1}_p(l) \cdot x_p \tag{91}$$

Using this, we can compute all of the optimality cuts in $O(m)$ time. Moreover, this makes it possible to omit using a nonlinear optimizer altogether. Thus, we will only need to use a linear programming optimizer to solve the series of master problems.

B. Starting point selection

In general, a starting point selection can greatly influence the convergence time of iterative procedures [19]. We want our starting point v^1 to not only lead to a feasible subproblem solution, but also be close to the optimum itself. Since our problem has a twice-differentiable function, we propose to produce the starting solution from the original problem, but with the function approximated by a Maclaurin series of order 1 and 2, that is:

$$\text{Linear start:} \quad \sum_{l\in L}\frac{f_l}{C_l+\varepsilon} \tag{92}$$

$$\text{Quadratic start:} \quad \sum_{l\in L}\Big(\frac{f_l}{C_l+\varepsilon}+\frac{f_l^2}{(C_l+\varepsilon)^2}\Big) \tag{93}$$

The linear approximation leads to the LP problem, and the quadratic approximation to the QP problem with a positive definite Hessian matrix; therefore, the starting point selection is a computationally easy procedure.

V. Numerical experiments

We have selected the network model with routing bottlenecks from the original article [6], which is shown in Fig. 1. On top of the picture, we see producers 1-5 who have to send their commodities to customers 6-10 through two bottlenecks represented by links 21 and 22. As this network

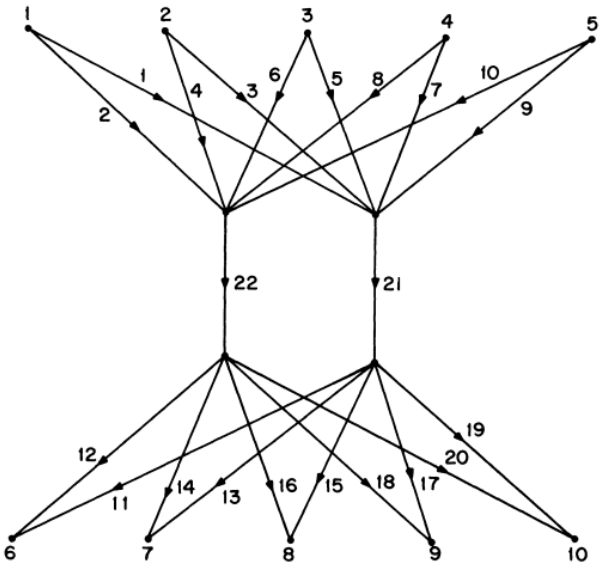

Fig. 1. Network graph, based on [6]. The vertices 1-5 represent heterogeneous suppliers, and the vertices 6-10 represent the recipients, which have demands for different suppliers' products. The arcs 21 and 22 constitute the network's bottlenecks.

is too small to argue for the usefulness of the decomposition, the tests were run on artificially constructed, larger network, but still preserving the "shape" of the original network with parameter values generated as such:

$$r_{\{p,c\}}=\frac{p\cdot c}{p_{\max}\cdot c_{\max}} \qquad \forall c\in\{1..c_{\max}\}, p\in\{1..p_{\max}\} \tag{94}$$

$$C_{\{p,n\}}=20\frac{\sum_{c\in C} r_{\{p,c\}}}{n\cdot(n_{\max}+1-n)} \qquad \forall p\in\{1..p_{\max}\}, n\in\{1..n_{\max}\} \tag{95}$$

$$C_{\{n\}}=30\frac{\sum_{p\in P}\sum_{c\in C} r_{\{p,c\}}}{n\cdot(n_{\max}+1-n)} \qquad \forall n\in\{1..n_{\max}\} \tag{96}$$

$$C_{\{n,c\}}=20\frac{\sum_{p\in P} r_{\{p,c\}}}{n\cdot(n_{\max}+1-n)} \qquad \forall c\in\{1..c_{\max}\}, n\in\{1..n_{\max}\} \tag{97}$$

where

- $p_{\max}, n_{\max}, c_{\max}$ - number of producers, bottlenecks and customers, respectively,
- $r_{\{p,c\}}$ - demand for customer-producer pair,
- $C_{\{p,n\}}, C_{\{n\}}, C_{\{n,c\}}$ - capacities of link from producer p to bottleneck n, bottleneck n link, and from bottleneck n to customer c, respectively.

The models were implemented in AMPL. We have run the GBD algorithm using Xpress 9.4.2 solver. We have compared the performance of Benders algorithm against commercial nonlinear solvers (without decomposition) – Baron 24.5.8 and Minos 5.51. The accuracy parameter in GBD was set to $\beta = 10^{-3}$, and the small constant was set to $\varepsilon = 10^{-6}$. The objective cuts were split, and feasibility cuts precomputed. Tests were run on a PC with the Linux Mint operating system, AMD Ryzen 7 3750H processor with Radeon Vega Mobile Gfx, and 16GB of RAM. The results are in the Table II. The numbers have been rounded to 3 decimal places.

TABLE II
Results for three problem sizes. The first column specifies the appropriate values of $(p_{\max}, n_{\max}, c_{\max})$. The objective and time values were rounded to 3 decimal places.

$(p,n,c)_{\max}$	Method	Solver	Objective	Time [s]
(100, 5, 100)	No decomposition	Minos	48.182	37.345
	No decomposition	Baron	48.182	10.786
	Benders–linear start	Xpress	48.182	130.535
	Benders–quadratic start	Xpress	48.182	4.100
(200, 5, 200)	No decomposition	Minos	94.935	590.914
	No decomposition	Baron	94.935	530.014
	Benders–linear start	Xpress	94.935	917.105
	Benders–quadratic start	Xpress	94.935	20.056
(200, 5, 400)	No decomposition	Minos	131.299	2411.695
	No decomposition	Baron	131.299	146.343
	Benders–linear start	Xpress	131.299	1661.551
	Benders–quadratic start	Xpress	131.299	51.251

As we see, the Benders procedure with linear start achieved worse times than Minos on small problems, but would take the lead when the size of the problem increased. However, Baron outperformed both Minos and linear start GBD in all cases. The GBD with a quadratic approximation function start would perform significantly better in all cases, which shows just how

much the correct starting point can speed the optimization procedure.

VI. Conclusions

Efficient routing is essential for sustaining high quality of service, especially considering the scale of modern Internet infrastructure. Unfortunately, the technological complexity and interactions between different network flows exhibit properties that make our models harder to solve, e.g., the nonlinear phenomena in queue-based models. In order to address those issues, we can apply decomposition techniques, such as Generalized Benders Decomposition (GBD), that can help us isolate the nonlinear model parts from purely affine ones, thus possibly changing the problem's class.

In our paper we present one such example, when we apply the GBD method to a routing problem with a nonlinear queueing delay function, obtaining a linear programming master's problem and multiple independent nonlinear subproblems.

First, we provide theoretical considerations for the GBD method:

- (T1) We propose that in the case of multiple primal subproblems, all of the feasibility and optimality cuts can be split and added independently of each other
- (T2) We extend the GBD convergence criteria to include the case where some of the independent subproblems have single-dimensional constraints, binding the master's and subproblem variables

Application of those propositions in our case enables us to omit the use of nonlinear optimization software altogether. Moreover, we present two options – linear and quadratic – to generate the starting point.

We provide computational insight into our proposition's performance by comparing the decomposed problems with both types of starting point selection with the performance of commercial nonlinear programming software applied to the problem without decomposition. The experiments were carried out on 3 networks of multiple sizes.

Our main findings are as follows:

- (F1) The obtained results show that the GBD managed to find the optimum, showing that GBD, after our improvements is, indeed, applicable to routing problems, and can change to problem's class to LP
- (F2) When linear starting point selection was applied, the runtimes could not compete with those of commercial solvers, especially on problems with smaller sizes
- (F3) The use of quadratic starting point selection leads to significantly lower runtimes than the commercial solvers, suggesting how crucial the starting point selection for GBD is.

To summarize, the GBD can be applied to change the problem's class altogether, which can enable us to solve complex routing problems. Some of those problems have multiple independent subproblems, for which we provide theoretical improvements (T1, T2). We apply the method to one such nonlinear problem, and as a result, we obtain a purely linear iterative problem (F1). However, such a decomposition might not be competitive against the commercial solvers (F2). Therefore, an appropriate starting point selection might be needed to achieve a better performance (F3).

References

[1] K. Salimifard and S. Bigharaz, "The multicommodity network flow problem: state of the art classification, applications, and solution methods," *Operational Research*, vol. 22, no. 1, pp. 1–47, Mar. 2022. [Online]. Available: https://doi.org/10.1007/s12351-020-00564-8

[2] M. Minoux, "Multicommodity Network Flow Models and Algorithms in Telecommunications," in *Handbook of Optimization in Telecommunications*, M. G. C. Resende and P. M. Pardalos, Eds. Boston, MA: Springer US, 2006, pp. 163–184. [Online]. Available: https://doi.org/10.1007/978-0-387-30165-5_7

[3] A. van den Nouweland, P. Borm, W. van Golstein Brouwers, R. Groot Bruinderink, and S. Tijs, "A Game Theoretic Approach to Problems in Telecommunication," *Management Science*, vol. 42, no. 2, pp. 294–303, Feb. 1996, publisher: INFORMS. [Online]. Available: https://pubsonline.informs.org/doi/abs/10.1287/mnsc.42.2.294

[4] M. Naserian and K. Tepe, "Game theoretic approach in routing protocol for wireless ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 3, pp. 569–578, May 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870508000966

[5] N. Hohn, D. Veitch, K. Papagiannaki, and C. Diot, "Bridging router performance and queuing theory," *SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 1, pp. 355–366, Jun. 2004. [Online]. Available: https://dl.acm.org/doi/10.1145/1012888.1005728

[6] E. M. Gafni and D. P. Bertsekas, "Two-metric projection methods for constrained optimization," *SIAM Journal on Control and Optimization*, vol. 22, no. 6, pp. 936–964, 1984. [Online]. Available: https://doi.org/10.1137/0322061

[7] L. Layuan, L. Chunlin, and Y. Peiyan, "Performance evaluation and simulations of routing protocols in ad hoc networks," *Computer Communications*, vol. 30, no. 8, pp. 1890–1898, Jun. 2007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S014036640700103X

[8] E. Amaldi, A. Capone, L. G. Gianoli, and L. Mascetti, "A MILP-Based Heuristic for Energy-Aware Traffic Engineering with Shortest Path Routing," in *Network Optimization*, J. Pahl, T. Reiners, and S. Voß, Eds. Berlin, Heidelberg: Springer, 2011, pp. 464–477.

[9] T. Larsson and N. Hedman, *Routing protocols in wireless ad-hoc networks : a simulation study*. Luleå University of Technology, 1998. [Online]. Available: https://urn.kb.se/resolve?urn=urn:nbn:se:ltu:diva-52142

[10] M. Fortin and R. Glowinski, "Chapter III On Decomposition-Coordination Methods Using an Augmented Lagrangian," in *Studies in Mathematics and Its Applications*, ser. Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems, M. Fortin and R. Glowinski, Eds. Elsevier, Jan. 1983, vol. 15, pp. 97–146. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168202408700286

[11] T. Larsson and D. Yuan, "An Augmented Lagrangian Algorithm for Large Scale Multicommodity Routing," *Computational Optimization and Applications*, vol. 27, no. 2, pp. 187–215, Feb. 2004. [Online]. Available: https://doi.org/10.1023/B:COAP.0000008652.29295.eb

[12] W. E. Wilhelm, "A Technical Review of Column Generation in Integer Programming," *Optimization and Engineering*, vol. 2, no. 2, pp. 159–200, Jun. 2001. [Online]. Available: https://doi.org/10.1023/A:1013141227104

[13] G. Carello, I. Filippini, S. Gualandi, and F. Malucelli, "Scheduling and routing in wireless multi-hop networks by column generation," in *Conference: International Network Optimization Conference (INOC)*, Jan. 2007.

[14] D. A. Tarvin, R. K. Wood, and A. M. Newman, "Benders decomposition: Solving binary master problems by enumeration," *Operations Research Letters*, vol. 44, no. 1, pp. 80–85, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167637715001558

[15] A. M. Geoffrion, "Generalized Benders Decomposition," *Journal of Optimization Theory and Applications*, vol. 10, no. 4, pp. 237–260, 10 1972. [Online]. Available: https://doi.org/10.1007/BF00934810

[16] A. Karbowski, "Generalized Benders Decomposition method to solve big mixed-integer nonlinear optimization problems with convex objective and constraints functions," *Energies*, vol. 14, no. 20, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/20/6503

[17] M. K. Awad, Y. Rafique, and R. A. M'Hallah, "Energy-aware routing for software-defined networks with discrete link rates: A Benders decomposition-*based* heuristic approach," *Sustainable Computing: Informatics and Systems*, vol. 13, pp. 31–41, Mar. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210537916301251

[18] J. R. Birge and F. Louveaux, *Two-Stage Recourse Problems*. New York, NY: Springer New York, 2011, pp. 181–263. [Online]. Available: https://doi.org/10.1007/978-1-4614-0237-4_5

[19] A. Jiménez-Cordero, J. M. Morales, and S. Pineda, "Warm-starting constraint generation for mixed-integer optimization: A Machine Learning approach," *Knowledge-Based Systems*, vol. 253, p. 109570, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122007894