Received: 11 July 2019 | Revised: 7 November 2020 | Accepted: 19 January 2021

DOI: 10.1002/net.22026

RESEARCH ARTICLE

WILEY

The invisible-hand heuristic for origin-destination integer multicommodity network flows

Richard S. Barr | **Thomas McLoud**

Department of Engineering Management, Information, and Systems, Lyle School of Engineering, Southern Methodist University, Dallas, Texas, USA

Correspondence
Richard S. Barr, Department of Engineering Management, Information, and Systems, Lyle School of Engineering, Southern Methodist University, Dallas, TX 75275, USA.
Email: barr@smu.edu

Abstract

Origin-destination integer multicommodity flow problems differ from classic multicommodity models in that each commodity has one source and one sink, and each commodity must be routed along a single path. A new invisible-hand heuristic that mimics economic markets' behavior is presented and tested on large-scale telecommunications networks, with solution times two orders of magnitude faster than CPLEX's LP relaxation, more dramatic MIP ratios, and small solution value differences.

KEYWORDS

economics, heuristics, integer programming, logistics, multicommodity, network optimization, telecommunications

1 | INTRODUCTION

The general problem for origin-destination integer multicommodity network flows (ODIMCF) consists of a network with limited capacity on one or more arcs and several distinct, non-interchangeable commodities sharing this limited network capacity to satisfy their respective demands and supplies. Hence, all commodities have separate, structurally identical networks with upper bounds on the sum of flows across corresponding arcs. While this description is consistent with the classic minimum-cost MCF problem [2, 5, 7, 17, 33], ODIMCF has two differentiating aspects:

1. Each commodity has a single source (supply node) and a single sink (demand node).
2. The entire flow of each commodity must follow a single path from its source (origin) to its sink (destination).

This last requirement makes ODIMCF an integer programming problem. As the number of commodities increases, the size of an ODIMCF instance grows rapidly. This combination of large instances and integrality requirements increases the difficulty of solving these problems.

This research is motivated by the presence of large instances of ODIMCF models in practice, with hundreds of thousands of constraints and millions of binary variables, which may require quick or repeated solution. Even modest problem instances can challenge the effectiveness of current optimization methodologies. The combination of these issues serves as a strong motivator for the development of more efficient solution techniques in terms of speed and solution quality.

This paper develops a new heuristic approach for the solution of ODIMCF problems. The algorithm has polynomial asymptotic bounds for both space and time. The minimal space requirement enables the solution of large problem instances for which testing demonstrates extremely small running times and near-optimal solutions.

2 | APPLICATIONS AND LITERATURE REVIEW

Large instances of ODIMCF occur in communications, package distribution, computer, transportation, supply-chain distribution, and traffic networks [1, 6, 7]. In a transportation example, Huntley et al. [16] describe a problem from the railroad industry:

TABLE 1 Mapping of applications to ODIMCF

Application	MPLS	Grain-car movement
Commodities	LSPs	Blocks
Demand	LSP bandwidth	Block length
Nodes	Switches and routers	Train arrival or departure at a station
Arcs	Network links	Remaining at a station or movement by train
Arc capacities	Link bandwidth	Maximum train length and station capacity

the movement of loaded grain cars that are grouped into blocks and moved from their origins to their destinations. The grain trains connecting stations have load limits on total weight and length and multiple blocks can share a train's capacity. The combination of a station and train arrival/departure times forms network nodes and blocks traverse across arcs representing track usage or waiting at a station. These nodes, arcs, and blocks form an ODIMCF instance, with each block of freight cars treated as a separate commodity.

Traffic routing in multi-protocol label-switching (MPLS) and similar network technologies, such as segment routing [22], is an instance of ODIMCF in the telecommunications industry [15, 20, 28, 30]. A label-switched path (LSP) is established for groupings of traffic having the same origin and destination in an MPLS network. All traffic assigned to an LSP will follow the same path across the network, yet the LSPs share the limited network bandwidth, expressed as capacities on the arcs. Girish et al. [15] provide a formulation for MPLS traffic routing consistent with ODIMCF having LSPs serve as the commodities, along with additional formulations for specializations of this problem. The number of LSPs in even a small MPLS network can be large since at least one LSP may be required to connect each node to every other network node. For example, in a small 30-node network, 800 or 900 LSPs (commodities) are typical in practice.

Table 1 summarizes the mapping of application components to ODIMCF elements for both MPLS and grain-car movement applications. Other ODIMCF applications similar to that of MPLS routing include: wavelength-division multiplexing in optical networks without bifurcated flow [26], the virtual network embedding problem of mapping virtual communications networks with heterogeneous topologies onto physical networks [23], provisioning long-term private virtual circuits between customer endpoints on a large backbone network [28], and satellite payload configuration to optimize power usage while ensuring sufficient signal amplification for retransmission on the downlink [18].

While these ODIMCF problems can be formulated as generic integer programming models [10, 24, 34], realistic instances are challenging to solve with current software and specialized approaches are warranted. Specialized exact algorithms have been developed by Barnhart et al. [6], Park et al. [25], and Moura et al. [23] that use column-generation and branch-and-bound techniques to solve small instances of ODIMCF. These approaches use price-directive decomposition to solve the linear programming relaxations at the nodes in a branch-and-bound tree. Cutting planes are used at the nodes to improve the solutions found at each node.

But heuristic techniques have also been developed for these problems to enable the solution of larger problem instances. Early work by Huntley et al. [16] utilizes simulated annealing [14] to approximately solve an ODIMCF problem. Details of the procedure are incomplete, but good results are claimed. Laguna and Glover [19] use Tabu search for the related bandwidth-packing problem. Resende and Ribeiro [28] applied the GRASP metaheuristic [27] to route private virtual circuits through a backbone telecommunications network. Amiri et al. [3, 4], Rolland et al. [29], and recently Fortz et al. [12] present Lagrangian-relaxation-based heuristics for ODIMCF. And Brun et al. [9] develop an approximation heuristic inspired by game theory's Nash equilibrium.

The following sections present a mathematical statement of the problem and develop a new heuristic based on classic economic principles. Computational testing on large problem sets demonstrates the effectiveness of this approach.

3 | MATHEMATICAL FORMULATION

In formulating an ODIMCF problem, the network topology, arc capacities, and commodity information are assumed to be deterministic and given. Let $\mathbb{B} = \{0, 1\}$ be the set of binary numbers, $\mathbb{R}$ be the set of real numbers, $\mathbb{R}_+$ be the set of positive real numbers, and $\mathbb{Z}_{0+}$ be the set of non-negative integers.

Define K, N, and A to be the sets of commodities, nodes, and directed arcs, respectively. For directed arc $a \in A$, let $c_a \in \mathbb{R}_{0+}$ be the non-negative cost per unit of flow, $u_a \in \mathbb{R}_+$ be the capacity limit, and $i_a \in N$ ($j_a \in N$) be the tail (head) of the arc. The characteristics c_a, u_a, i_a, and j_a are universally associated with each $a \in A$.

Within the network, a *route*, $P \subset A$, is a set of arcs with the following characteristics:

1. If $P \neq \emptyset$, then P has an origin (destination) node $s \in N$ ($t \in N$) at which P originates (terminates).

TABLE 2 ODIMCF problem components

Component	Type	Definition
$T(n) \subset A$	Constant	Set of arcs terminating at $n \in N$
$E(n) \subset A$	Constant	Set of arcs emanating from $n \in N$
$u_a \in \mathbb{R}_+$	Constant	Capacity limit on total flow for $a \in A$
$c_a \in \mathbb{R}_{0+}$	Constant	Cost per unit of flow for $a \in A$
$i_a \in N$	Constant	Node from which $a \in A$ emanates
$j_a \in N$	Constant	Node at which $a \in A$ terminates
$s_k \in N$	Constant	Origin or source node for $k \in K$
$t_k \in N$	Constant	Destination or sink node for $k \in K$
$d_k \in \mathbb{R}_+$	Constant	Demand (supply) for commodity $k \in K$ at t_k (s_k)
$b_n^k \in \mathbb{B}$	Constant	$1 \Rightarrow n = t_k, -1 \Rightarrow n = s_k, 0$ otherwise. $k \in K, n \in N$
$\mathrm{X}_{a,k} \in \mathbb{B}$	Variable	1(0)$\Rightarrow$ commodity $k \in K$ uses (does not use) arc $a \in A$

2. $\forall a \in P, j_a \neq t \Rightarrow \exists b \in P$ s.t. $j_a = i_b$.
3. $\forall a \in P, i_a \neq s \Rightarrow \exists b \in P$ s.t. $j_b = i_a$.
4. $P \neq \emptyset \Rightarrow \exists a \in P$ s.t. $i_a = s\ (j_a = t)$.
5. $\forall a \in P,$ there does not exist $b \in P$ s.t. $i_a = i_b\ (j_a = j_b)$.
6. If $P \neq \emptyset$, then the directed network formed by the directed arcs of P and their heads and tails is a tree.

Each commodity $k \in K$ has an origin $s_k \in N$, destination $t_k \in N$, and required flow from s_k to t_k of $d_k \in \mathbb{R}_+$. This demand for commodity k is represented by d_k units of supply at s_k and d_k units of demand at t_k indicated in the demand vector $\mathbf{b}^k$ with a 1 (-1) entry corresponding to t_k (s_k) and 0 for all other nodes. The characteristics s_k, t_k, d_k, and $\mathbf{b}^k$ are universally associated with each $k \in K$.

Let $\mathbf{X}$ be a matrix of binary flow variables for all commodities. If $\mathrm{X}_{a,k}$ is 1 (0) then commodity k uses (does not use) arc $a \in A$. For node $n \in N$, let $E(n)$ be the set of directed arcs emanating from n and $T(n)$ be the set of directed arcs terminating at n. Table 2 contains a summary of the components of an ODIMCF problem.

The node-arc formulation for ODIMCF is given by (1)–(4).[1] The objective function, (1), seeks to minimize total routing cost for all commodities. The node-balance equations, (2), ensure that the flow of each commodity satisfies the conservation of flow at the nodes and supply and demand requirements. The limit on arc capacities is enforced across all commodities in (3). The integrality requirements, (4), require that the flows for each commodity follow a single path through the network.

$$
\begin{aligned}
&[\text{ODIMCF}] && && \\
&\text{Minimize:} && \sum_{k\in K}\sum_{a\in A} d_k c_a \mathrm{X}_{a,k} && && (1)\\
&\text{subject to:} && \sum_{a\in T(n)} \mathrm{X}_{a,k} - \sum_{a\in E(n)} \mathrm{X}_{a,k} = b_n^k && \forall n \in N,\ \forall k \in K, && (2)\\
& && \sum_{k\in K} d_k \mathrm{X}_{a,k} \leq u_a && \forall a \in A, && (3)\\
& && \mathrm{X}_{a,k} \in \mathbb{B} && \forall k \in K,\ \forall a \in A. && (4)
\end{aligned}
$$

4 | INVISIBLE-HAND HEURISTIC FOR ODIMCF

The solution of large instances of ODIMCF have proven to be challenging for standard optimization techniques [6]. With this as motivation, a new heuristic is developed that quickly determines near-optimal solutions for large-scale problems with many commodities.

Many successful metaheuristics are inspired by systems that evolved naturally. Corne et al. [11] and Gendreau and Potvin [14] present many examples of such approaches, which include genetic algorithms, immune-system methods, ant-colony optimization, and particle swarm. Garlick and Barr [13] use ant-colony optimization for the routing and wavelength assignment problem, which has many characteristics in common with ODIMCF.

[1]For a path-based formulation, see [6,16].

The new heuristic presented below is inspired by Adam Smith's insights into market-based economic systems. In 1776, Adam Smith wrote the following [31]:

> Every individual necessarily labours to render the annual revenue of the society as great as he can. He generally neither intends to promote the public interest, nor knows how much he is promoting it … He intends only his own gain, and he is in this, as in many other cases, led by an *invisible hand* to promote an end which was no part of his intention. Nor is it always the worse for society that it was no part of his intention. By pursuing his own interest he frequently promotes that of the society more effectually than when he really intends to promote it. I have never known much good done by those who affected to trade for the public good.

Based on Smith's observation, the *invisible-hand heuristic* (IHH) is designed to emulate and exploit the forces at work in a competitive marketplace. A specific application of this approach is developed for ODIMCF, wherein each commodity must choose a path over which to be routed. Just as the price mechanism is the control mechanism of a true market system [8, 21, 32], IHH uses resource prices as its control mechanism, where the resources are the arc capacities. IHH's pricing mechanism consists of two components, the original arc costs and a heuristic *scarcity cost* unique to each arc. The *market cost* of an arc is the sum of scarcity cost and the original arc cost. The original arc costs are infinitely elastic, not varying with quantity of an arc's capacity consumed by commodities. The scarcity cost function is designed to become increasingly inelastic as the quantity of an arc's capacity is consumed—for each additional unit of capacity consumed the slope of the marginal cost function increases. The increasing resource price works with the commodity demand curves' rationing function to help limit consumption of the scarce resources in the network—arc capacity. In this way, each arc is an independent monopolistic supplier of a unique resource and adjusts the price of it based solely on the current demand it sees for its capacity.

The commodities in the ODIMCF problem are the consumers of the resources with each trying to acquire a set of complementary goods—capacity on specific arcs—to form a path from its origin to its destination that minimizes the total market cost of the path. For a commodity, the arcs along a possible path from origin to destination are complementary goods so that the price of capacity on one arc affects the commodity's demand for capacity on other arcs in the path. As the price for an arc's capacity goes down (up), the commodity's demand for capacity on complementary arcs will go up (down). As each commodity has multiple paths to choose from in the network, arc capacity for arcs in alternate paths are substitute goods. As the price for capacity on an arc goes down (up), the commodity's demand for capacity on substitute arcs will go down (up). As each commodity may have a different origin-destination pair with different possible paths, the set of complementary and substitute goods will vary by commodity.

IHH does not attempt to determine the demand curve for each arc's capacity. Nor does IHH have a central planner coordinating individual arc prices or allocating arc capacity to specific commodities. Instead, each commodity continuously attempts to minimize the market cost of its route as the costs change. This process of continuous reevaluation proceeds until an equilibrium is reached and all commodities are satisfied with their routes. The commodities never consider the effect of routing decisions on other commodities (the entire society); each commodity considers only its own self-interest. The only interaction between commodities and between arcs and commodities occurs through the price mechanism.

4.1 | Residual capacity and market costs

ODIMCF problems have hard limits on the availability of each resource—arc's capacity—and is a short-run problem where no additional capacity can be added. To satisfy the arc-capacity limits, the marginal market cost curve is designed to reach an equilibrium point where the total capacity utilized by the commodities is at most the available supply. As ODIMCF is also trying to minimize total routing cost and not merely satisfy the capacity constraints, the marginal market cost curve must also reflect the original arc costs.

The scarcity cost component of the marginal market cost is focused on achieving equilibrium and follows the law of diminishing returns with marginal cost rising as utilized capacity for an arc approaches the arc's capacity limit, u_a. The scarcity cost of an arc varies by commodity and is a function of the residual capacity of the arc and the d_k for the commodity k. As defined in Table 3, let $r(a, k, \mathbf{X}) \in \mathbb{R}$ (5) be the *residual capacity* available for commodity $k \in K$ on arc $a \in A$, with no requirement that it be non-negative. The scarcity cost function $sc(a, k, \mathbf{X}) \in \mathbb{R}$ (6) reflects an increase in cost or price for commodity k as residual capacity on arc a approaches zero. The associated parameters β, μ, and π are positive real-valued scalars, each having the same value for all arcs and commodities. The values of these parameters are set a priori and are discussed further in Section 4.2.

The marginal market cost of commodity k on arc a is determined by the function $mc(a, k, \mathbf{X}) \in \mathbb{R}_{0+}$ (7) and is a marginal cost in that it represents the cost of the last unit of flow if commodity k were to use arc a. Function $rc(P, k, A, \mathbf{X}) \in \mathbb{R}_{0+}$ (8) defines the *marginal market cost* of route $P \subset A$.

TABLE 3 Market and scarcity cost definitions

$r(a,k,\mathbf{X}) = u_a - \sum_{g\in K\setminus\{k\}} d_g X_{a,g}$	(5)
$sc(a,k,\mathbf{X}) = \mu\left(max\left(0, \frac{\beta+d_k-r(a,k,\mathbf{X})}{\beta}\right)\right)^{\pi}$	(6)
$mc(a,k,\mathbf{X}) = sc(a,k,\mathbf{X}) + c_a$	(7)
$rc(P,k,A,\mathbf{X}) = \sum_{a\in P} mc(a,k,\mathbf{X})$	(8)

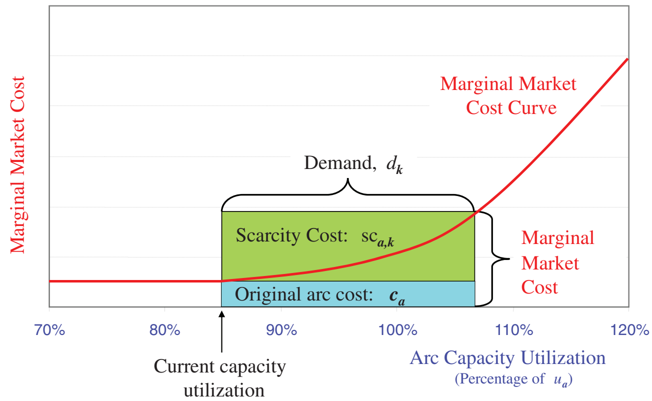

FIGURE 1 Marginal market cost curve [Color figure can be viewed at wileyonlinelibrary.com]

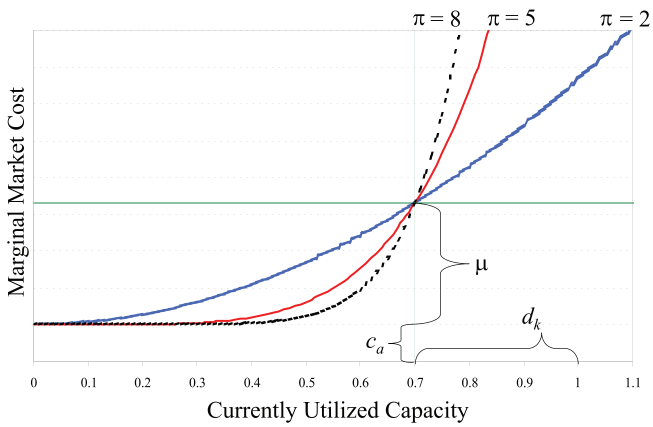

FIGURE 2 Marginal market cost curve with alternative π values [Color figure can be viewed at wileyonlinelibrary.com]

Since $r(a,k,\mathbf{X})$ changes based on the current route selection of the other commodities, $K\setminus\{k\}$, the marginal market cost of an arc is not fixed and varies as the flows of other commodities change. As $sc(a,k,\mathbf{X})$ is dependent upon commodity k's demand, d_k, market costs vary by commodity. Figure 1 shows a graphical representation of $mc(a,k,\mathbf{X})$ as it relates to $r(a,k,\mathbf{X})$ and d_k. In the diagram, the current utilization corresponds to $u_a - r(a,k,\mathbf{X})$, the arc capacity utilized by other commodities. The market cost is determined as the intersection of the resulting arc capacity utilization if commodity k uses arc a, $u_a - r(a,k,\mathbf{X}) + d_k$, and the marginal market cost curve.

4.2 | Scarcity cost parameters and the market cost curve

The parameters β, μ, and π control the shape of the marginal market cost curve and determine at what arc capacity utilization the market cost is no longer infinitely elastic and the rate at which the market cost becomes increasingly inelastic. These effects are visible in the market cost curve, the plot of market cost for arc $a\in A$ against currently allocated arc capacity with respect to commodity $k\in K$, $u_a - r(a,k,\mathbf{X})$. Increasing π affects the rate of change in the slope of the curve—how fast the price becomes inelastic. Increasing π results in a decrease in marginal market cost for the region $d_k < r(a,k,\mathbf{X})$ and an increase for the region $d_k > r(a,k,\mathbf{X})$. The region $d_k \le r(a,k,\mathbf{X})$ corresponds to a set of flows for which commodity k can be routed on arc a without violating the capacity constraint for a, u_a. Decreasing π has the opposite effect. Figure 2 shows the marginal market cost curve with three different values of π with all other parameters held constant.

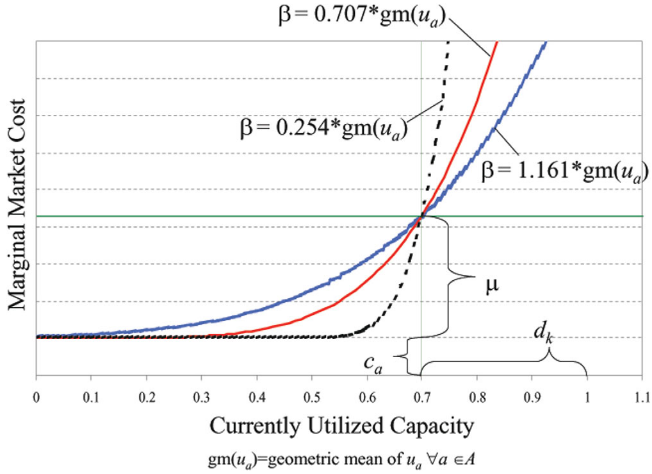

FIGURE 3 Marginal market cost curve with alternative β values [Color figure can be viewed at wileyonlinelibrary.com]

The parameter β determines the point at which the scarcity cost component of the market cost becomes nonzero and the marginal market cost is not infinitely elastic as $sc(a, k, \mathbf{X})$ becomes nonzero when $r(a, k, \mathbf{X}) < \beta + d_k$. A secondary effect is that β affects the slope of the curve as the scarcity cost rises from 0 to μ over a change of β in $(u_a - r(a, k, \mathbf{X}))$. Figure 3 illustrates three alternative values of β.

Finally, parameter μ controls the magnitude of $sc(a, k, \mathbf{X})$ in a linear manner. Allocated capacity values below zero are not shown for any cost curve as $r(a, k, \mathbf{X}) \leq u_a \Rightarrow (u_a - r(a, k, \mathbf{X})) \geq 0$. The parameters β, μ, and π may be used to manipulate the shape of the market cost curve to adjust for different applications. Section 5.2 describes one method for adjusting the parameters for a specific application.

4.3 | IHH algorithmic steps

The IHHO($\mathcal{P}$) heuristic for ODIMCF is given in Algorithm 1, where $\mathcal{P} = (N, A, K, \mathbf{X}, \beta, \mu, \pi)$ represents an ODIMCF problem, the current values for the decision variables, and the values of the scarcity-cost parameters.

Algorithm 1. IHHO($\mathcal{P}$) algorithm

Input: $\mathcal{P}$
Output: $\mathbf{X}$
$\mathbf{X}$ ←SPSolve($\mathcal{P}$) // Routecommodities on minimum original arc cost paths.
if Feasible($A, K, \mathbf{X}$) = TRUE **then** // Is trivial SPSolve($\mathcal{P}$) solution feasible?
  Stop
end if
more ← TRUE // Boolean variablecontrolling termination of main loop.
$\lambda_k \leftarrow 0, \forall k \in K$ // Setrouting change counts to 0.
while *more* = TRUE **do** // Main loop.
  more ← FALSE
  Randomize order of K
  for all $k \in K$ **do** // All commodities reexamine routing decision in randomorder.
    if Route($k, \lambda_k, \mathcal{P}$) = TRUE **then** // Does k's routing decision change?
      $\lambda_k \leftarrow \lambda_k + 1$ // Incrementrouting change counter.
      more ← TRUE // Market costs maybe altered for $g \in K \setminus \{k\}$.
    end if
  end for
end while
$\mathbf{X}$ ←ResidualCapacityClearance($\mathcal{P}$) // Reduce arc costs to original if excess capacity exists
Return $\mathbf{X}$

The IHHO($\mathcal{P}$) algorithm starts with an initial solution found by the SPSolve($\mathcal{P}$) procedure, which routes each commodity, k, on the shortest path from s_k to t_k based on the original arc costs without regard to arc capacities (Algorithm 2). This is done with the SP(k, N, A) algorithm for the shortest-path problem defined in (9)–(11), where $\mathbf{x}$ is a vector of flow variables for the shortest path found. SP(k, N, A) returns the set of $s_k \to t_k$ path arcs or $\emptyset$ if no path is found.

$$[\mathrm{SP}(k)]$$

$$\text{Minimize:} \quad \sum_{a \in A} c_a x_a \tag{9}$$

$$\text{subject to:} \quad \sum_{a \in T(n)} x_a - \sum_{a \in E(n)} x_a = b_n^k \qquad \forall n \in N,\ \forall k \in K, \tag{10}$$

$$x_a \in \mathbb{B} \qquad \forall a \in A. \tag{11}$$

Algorithm 2. SPSolve($\mathcal{P}$) procedure

Input: $\mathcal{P}$
Output: $\mathbf{X}$
1: **for all** $k \in K$ **do**
2:   $NewPath \leftarrow$ SP(k, N, A) // Find a shortestpath from s_k to t_k using original arc costs.
3:   **if** $NewPath \neq \emptyset$ **then** // Does a path from s_k to t_k exist?
4:     $\mathrm{X}_{a,k} \leftarrow 1,\ \forall a \in NewPath$
5:     $\mathrm{X}_{a,k} \leftarrow 0,\ \forall a \in A \setminus NewPath$
6:   **end if**
7: **end for**
8: Return $\mathbf{X}$

This initial solution from SPSolve($\mathcal{P}$) is checked for feasibility with respect to the arc-capacity constraints by procedure Feasible$(A, K, \mathbf{X})$ (not shown). The solution is expected to be infeasible; if this trivial solution is feasible, IHHO($\mathcal{P}$) returns it as the optimal solution and exits.

After the initial solution is found in IHHO($\mathcal{P}$), the variables *more* and λ_k are initialized. The Boolean variable *more* controls the termination of the main loop. The variable $\lambda_k \in \mathbb{Z}_{0+}$ is the count of routing changes for commodity k and is used to make a routing decision in the Route$(k, \lambda_k, \mathcal{P})$ procedure, described in the next section. IHHO($\mathcal{P}$) then iteratively reevaluates the routing of each commodity until an equilibrium is reached and all commodities are satisfied with their current routing decisions based upon current marginal market costs. This state is indicated when *more* is FALSE.

During every iteration, each commodity reexamines its routing based upon current market costs, $mc(a, k, \mathbf{X})$, using the Route$(k, \lambda_k, \mathcal{P})$ procedure. The order in which commodities reevaluate their routing decisions is random. Each commodity examines its decision once per iteration. This ordering of commodities is implemented to avoid giving bias or preferential treatment toward any single commodity or group of commodities. (If a preference for some commodities is desirable, the ordering can be altered to reflect that bias.)

A change in routing for commodity $k \in K$ is indicated by the results of the Route$(k, \lambda_k, \mathcal{P})$ procedure: FALSE (TRUE) indicates no change (a change). If no change occurred, the total flow and market costs for all arcs for all commodities are also unchanged. If a reroute of k is indicated, then the market costs for two or more arcs may have also changed for all other commodities. Commodities having already evaluated their routing decisions before k during the current iteration of the main loop will require the opportunity to reevaluate their decisions based on the new market costs. This requirement is indicated by setting *more* to TRUE and satisfied by executing a subsequent iteration.

The final step in IHHO($\mathcal{P}$) is the use of the ResidualCapacityClearance($\mathcal{P}$) procedure, detailed in Section 4.3.2 to search for lower-cost routes based on the original arc costs. This process only reassigns commodities to feasible routes: routes with sufficient residual capacity.

4.3.1 | Routing decision

Algorithm 3 shows the Route$(k, \lambda_k, \mathcal{P})$ procedure for reexamining the routing decision of commodity $k \in K$ and adjusting the associated decision variables. Let $CP(k) = \{a \in A : \ \mathrm{X}_{a,k} = 1\}$ be the set of directed arcs currently used by commodity $k \in K$. Commodity k makes a routing decision by considering the marginal market cost, the combination of scarcity cost and original

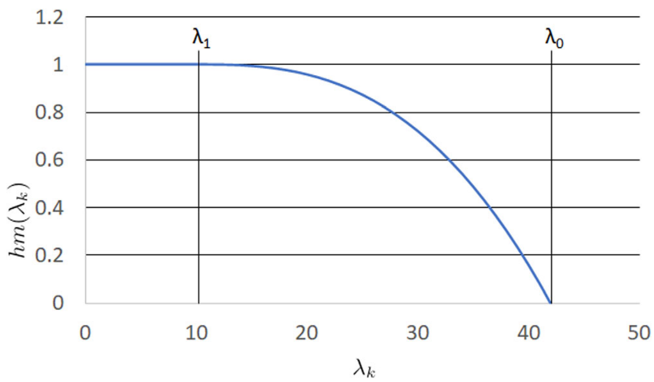

FIGURE 4 Hurdle multiplier curve for λ_1 and λ_0 [Color figure can be viewed at wileyonlinelibrary.com]

arc cost, in finding a route from its origin, s_k, to its destination, t_k. Route(k, λ_k, $\mathcal{P}$) determines a new route, *NewPath* $\subset A$, based on market costs, which is then compared with $CP(k)$, the incumbent best route for commodity k, to determine if it is a new best route.

Algorithm 3. Route(k, λ_k, $\mathcal{P}$) procedure

Input: $k \in K$, $\lambda_k \in \mathbb{Z}_{0+}$, $\mathcal{P}$
Output: TRUE if $k \in K$ has changed routing, else FALSE
NewPath $\leftarrow$ SPS($k, \mathcal{P}$)// Find the shortest $s_k - t_k$ path based on market costs.
if *NewPath* $\neq \emptyset$ **then**
  if $rc(NewPath, k, A, \mathbf{X}) < hm(\lambda_k) rc(CP(k), k, A, \mathbf{X})$ **then** // Has an improved new route beenfound?
    // Update decision variables.
    $\mathrm{X}_{a,k} \leftarrow 0$, $\forall a \in CP(k)$
    $\mathrm{X}_{a,k} \leftarrow 1$, $\forall a \in NewPath$
    Return TRUE // Routing decision has changed.
  end if
end if
Return FALSE // Routing decision remains the same.

NewPath is found using SPS($k, \mathcal{P}$), which finds the minimum cost path from s_k to t_k based on the current market cost for commodity $k \in K$, $mc(a, k, \mathbf{X})$. SPS($k, \mathcal{P}$) solves the problem in (12)–(14), where $\mathbf{x}$ is a vector of binary flow variables determining *NewPath*. SPS($k, \mathcal{P}$) returns the set of arcs in the shortest path discovered or Ø if no path is found.

$$[\mathrm{SPS}(k)]$$

$$\text{Minimize:} \quad \sum_{a \in A} mc(a, k, \mathbf{X}) x_a \tag{12}$$

$$\text{subject to:} \quad \sum_{a \in T(n)} x_a - \sum_{a \in E(n)} x_a = b_n^k \qquad \forall n \in N \tag{13}$$

$$x_a \in \mathbb{B} \qquad \forall a \in A \tag{14}$$

The value of the variable λ_k passed to Route(k, λ_k, $\mathcal{P}$) by IHHO($\mathcal{P}$) is the number of times commodity $k \in K$ has changed routing. Let the cost hurdle multiplier, $hm(\lambda_k) \in \mathbb{R}$, be a (user-defined) monotonically decreasing function in the range [0,1] such that $\lambda_k \in \mathbb{Z}_{0+}$ and $\exists \lambda_0 < \infty$ s.t. $hm(\lambda_0) = 0$. The new route, *NewPath*, replaces the incumbent route, $CP(k)$, only if $rc(NewPath, k, A, \mathbf{X}) < hm(\lambda_k) rc(CP(k), k, A, \mathbf{X})$. *NewPath* must provide a certain level of improvement over the incumbent with respect to current market costs for a replacement to occur. As commodity k changes routes more regularly, λ_k will increase and $hm(\lambda_k)$ will decrease, as shown in Figure 4. The gradual decrease in the cost hurdle multiplier requires subsequent *NewPath*s to provide an increasingly substantial improvement over $CP(k)$. To allow IHHO($\mathcal{P}$) to achieve an equilibrium without being artificially forced into the equilibrium, $hm(\lambda_k)$ is designed to have a value of 1 until $\lambda_k \geq \lambda_1$. With $hm(\lambda_k) = 1$, routes *NewPath* and $CP(k)$ are compared based solely on their marginal market costs.

With other commodities changing routes, $rc(CP(k), k, A, \mathbf{X})$ may change between iterations even if $CP(k)$ is the same. If this path's residual capacity decreases significantly and the current $CP(k)$ becomes untenable, $hm(\lambda_k) rc(CP(k), k, A, \mathbf{X})$ will increase, possibly enabling a previously rejected route to replace the incumbent.

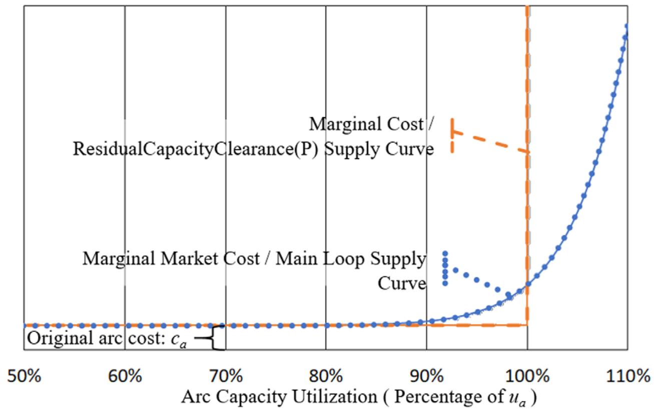

FIGURE 5 Main loop supply curve vs. ResidualCapacityClearance($\mathcal{P}$) supply curve [Color figure can be viewed at wileyonlinelibrary.com]

Creating $hm(\lambda_k)$ such that $hm(\lambda_0) = 0$ for some $\lambda_0 < \infty$ ensures each commodity can change routes at most λ_0 times, as no *NewPath* can have a cost less than 0. If IHHO($\mathcal{P}$) is unable to reach an equilibrium, this limit on the number of changes per commodity ensures the termination of IHHO($\mathcal{P}$).

4.3.2 | ResidualCapacityClearance($\mathcal{P}$) Procedure

The scarcity cost component of the marginal market cost reflects arcs charging what the market will bear for high-demand, limited resources. However, an arc's actual marginal cost for arc capacity is the original arc cost, c_a. At the equilibrium reached at the end of the main loop of the IHHO($\mathcal{P}$) algorithm, some arcs will be left with excess capacity because no commodities (i.e., consumers) were willing to pay the marginal market cost for the unused capacity. During the ResidualCapacityClearance($\mathcal{P}$) procedure each arc will attempt to sell its unused capacity at the marginal cost to induce consumers to switch from substitute goods (alternate paths). An arc will not offer a commodity capacity above the arc's upper limit, u_a, unless the commodity had paid the marginal market cost and was routed over the arc at the end of the main loop. The new supply curve will be infinitely elastic up to full arc-capacity utilization and infinitely inelastic once the arc's capacity is fully utilized (per Figure 5).

The ResidualCapacityClearance($\mathcal{P}$) procedure, Algorithm 4, examines the commodities to find routes incurring lower original-arc cost while not violating arc capacity constraints. If an improved route is found, $CP(k)$ is switched to the new path. If an improved route is not found, $CP(k)$ is left as is. If $CP(k)$ is not altered by ResidualCapacityClearance($\mathcal{P}$), $CP(k)$ may contain arcs whose capacity constraints are violated. ResidualCapacityClearance($\mathcal{P}$) enforces capacity constraints only for altered routes. ResidualCapacityClearance($\mathcal{P}$) does not un-route routed commodities; $CP(k) = \emptyset$ after ResidualCapacityClearance($\mathcal{P}$) only if $CP(k) = \emptyset$ at the start of ResidualCapacityClearance($\mathcal{P}$).

Algorithm 4. ResidualCapacityClearance($\mathcal{P}$) procedure

---

Input: $\mathcal{P}$
Output: $\mathbf{X}$

$\lambda_k \leftarrow 0, \forall k \in K$
more $\leftarrow$ TRUE
while *more* = TRUE **do**
  more $\leftarrow$ FALSE
  Randomize order of K
  for all $k \in K$ **do**
    NewPath $\leftarrow$ SPM($k, \mathcal{P}$)
    if $omrc(NewPath, k, K\mathbf{X}) < min(\mathrm{M}, hm(\lambda_k) omrc(CP(k), k, \mathbf{X}))$
    then
      $\lambda_k \leftarrow \lambda_k + 1$
      $\mathrm{X}_{a,k} \leftarrow 0, \forall a \in CP(k)$
      $\mathrm{X}_{a,k} \leftarrow 1, \forall a \in NewPath$
      more $\leftarrow$ TRUE
    end if
  end for
end while
Return $\mathbf{X}$

The *original marginal cost*, $omc(a,k,K,\mathbf{X}) \in \mathbb{R}_{0+}$ (15), is given as the original arc cost, c_a, if arc a has at least d_k residual capacity, otherwise infinity. These are used to determine the *original marginal route cost* of route P for commodity k, $omrc(P,k,\mathbf{X}) \in \mathbb{R}_{0+}$ (16).

$$omc(a,k,K,\mathbf{X}) = \begin{cases} c_a & \text{if } \; r(a,k,\mathbf{X}) - d_k \geq 0, \\ \mathrm{M} & \text{otherwise.} \end{cases} \tag{15}$$

$$omrc(P,k,\mathbf{X}) = \sum_{a\in P} omc(a,k,K,\mathbf{X}) \tag{16}$$

where scalar $\mathrm{M} \geq max(\mathbf{c})\,|N|$.

$$\left[\mathrm{SPM}(k)\right]$$

$$\text{Minimize:} \quad \sum_{a\in A} omc(a,k,K,\mathbf{X})x_a \tag{17}$$

$$\text{subject to:} \quad \sum_{a\in T(n)} x_a - \sum_{a\in E(n)} x_a = b_n^k \qquad \forall n \in N \tag{18}$$

$$x_a \in \mathbb{B} \qquad \forall a \in A \tag{19}$$

ResidualCapacityClearance($\mathcal{P}$) solves SPM($k,\mathcal{P}$) for each commodity $k \in K$ having its routing reexamined. The formulation for the problem solved is shown in Equations (17)–(19). SPM($k,\mathcal{P}$) returns the set of arcs in the shortest path based on feasible arc costs from s_k to t_k. If $omrc(NewPath,k,\mathbf{X}) < \mathrm{M}$, then this is the shortest path with respect to the original arc costs on which sufficient residual arc capacity exists to route commodity k. If $omrc(NewPath,k,\mathbf{X}) \geq \mathrm{M}$, then there does not exist a path from s_k to t_k with at least d_k residual capacity on each arc and no feasible path exists.

4.4 | Interpretation of IHH's final solution

The final solution found by IHHO($\mathcal{P}$) provides a single route for each commodity through the network and will always meet the ODIMCF node-balance and integrality constraints (2) and (4). If the final solution is feasible with respect to the arc capacity constraints (3), then an integer feasible solution is at hand. Unlike some integer programming techniques, IHHO($\mathcal{P}$) does not provide a bound on the optimality gap, the difference between the objective function values of the optimal solution and the IHHO($\mathcal{P}$) solution. Other techniques often rely on the linear programming relaxation as a source of gap information. While this relaxation of ODIMCF can be solved to provide such gap information, testing indicates the time required to solve the relaxation will be greater than the time required by IHHO($\mathcal{P}$) to find an integer feasible solution.

If the solution found by IHHO($\mathcal{P}$) violates one or more arc capacities, the solution will be infeasible, meaning no feasible solution to that particular ODIMCF problem exists or the heuristic could not identify such a solution. This situation was not encountered in the computational testing.

4.5 | Asymptotic bounds

IHHO($\mathcal{P}$) has polynomial asymptotic bounds with respect to both time and space. The asymptotic bound on space requirements is $O(|A| + |N||K|)$. The $|A|$ term represents the space needed to store arc information. The $|N||K|$ term represents the space required to store route information. As the assumption is made that arc costs are always non-negative, a route will contain at most $|N|-1$ arcs with no cycles. One route is stored for each commodity. The $O(|N|)$ space required for storage of node information is dominated by the $|N||K|$ term. Similarly, a $|K|$ term representing commodity information is omitted.

The asymptotic bound on running time for IHHO($\mathcal{P}$) is $O(\lambda_0|K|^2(|A| + |N|\log|N|))$. SP($k,N,A$), SPM($k,\mathcal{P}$), and SPS($k,\mathcal{P}$) use a shortest-path algorithm with a time bound of $O(|A| + |N|\log|N|)$, under the assumption of non-negative arc costs [2]. Route($k,\lambda_k,\mathcal{P}$) uses SPS($k,\mathcal{P}$) once and requires $O(|N|)$ additional time to update and compare routes for a time bound of $O(|A| + |N|\log|N|)$. SPSolve($\mathcal{P}$) uses SP(k,N,A) and records a route once for each commodity for a total of $O(|K|(|A| + |N|\log|N|))$ time. Each commodity can change routes at most λ_0 times before the cost-hurdle makes further change impossible. In the worst case, at most one commodity will change routes during each iteration of the main loop of IHHO($\mathcal{P}$). This worst case results in $\lambda_0|K|$ executions of the main loop requiring $O(\lambda_0|K|^2(|A| + |N|\log|N|))$ time. The main loop within ResidualCapacityClearance($\mathcal{P}$) is similar to the main loop of IHHO($\mathcal{P}$) resulting in a $O(\lambda_0|K|^2(|A| + |N|\log|N|))$ time bound for ResidualCapacityClearance($\mathcal{P}$).

TABLE 4 IHHO($\mathcal{P}$) parameters

Parameter	Value
β	$\max\left(\frac{\sqrt{2}}{2} gm\left(u_a\right), gm(d_k)\right)$
μ	$gm(c_a)\sqrt{e}$
π	e^e
λ_0	43
λ_1	10
$hm(\lambda_k)$	1, if $\lambda_k < \lambda_1$; else $1 - \left(\frac{\lambda_k - \lambda_1}{\lambda_0 - \lambda_1 - 1}\right)^e$

$gm(y_z)$ is the geometric mean of values for y_z $\forall z \in Z$.

5 | COMPUTATIONAL TESTING

Computational testing is designed to determine the performance characteristics of an IHHO($\mathcal{P}$) implementation and compare them to those of commercial-grade software. Test problems are generated to measure the responses of running time and solution quality to changes in network and commodity characteristics.

The reported running times do not include time to read the problem or record the solutions. Solution quality is compared with values for the LP relaxation (LPR) and the best-available integer programming (MIP) solution.

5.1 | Test environment

All benchmark testing is performed on a Dell R720 with dual Six Core Intel Xeon 3.5 GHz processors and 252 GB RAM at Southern Methodist University's Lyle School of Engineering. The IHHO($\mathcal{P}$) algorithm is implemented in C++ and compiled with g++ at the default optimization level. Reported running times (CPU execution times) are exclusive of input and output processing.

LPR and MIP solutions are generated using IBM ILOG CPLEX Interactive Optimizer 12.6.0.0 (CPLEX12). CPLEX12 is run with default settings, with three exceptions: the MIP time limit is set to 7200 s, the optimality tolerance increased to 0.25%, and single-thread mode is used.[2]

5.2 | Parameter selection

Performance of IHHO($\mathcal{P}$) is affected by the parameters β, μ, and π. To avoid manually varying these values to find a set of parameters with good performance over a range of problems, the metaheuristic differential evolution (DE) [11] was used to select their values, as follows. Three problems were generated for four groups with different problem dimensions. DE is an evolutionary algorithm with a population. For this application, a member of the population was defined by 3-tuple of values for β, μ, and π. DE evaluated each member of the population by running the problems through IHHO($\mathcal{P}$) with the member's β, μ, and π values and taking the mean of the routing cost normalized to known solution values. Members evaluated with lower cost were preferred during the creation of the next generation. A population size of 30 was used. Following 100 DE generations, the benchmark testing values for β, μ, and π were determined from the group results and are shown in Table 4. The parameters were tuned on a completely different set of problems from those used in the testing reported herein. In addition, the two sets of problems were generated using different problem generators for both the networks and commodities.

While DE used IHHO($\mathcal{P}$) as a subroutine to automatically determine a set of parameters, IHHO($\mathcal{P}$) has no dependency on DE. Other methods for tuning the parameters β, μ, and π could have been used. If IHHO($\mathcal{P}$) is to be used for a new application and similar benchmark problems are available, re-tuning the parameters is recommended. The use of an automated tuning method such as DE would facilitate periodic reevaluation of these parameters, and new problems could be added to the set of benchmark problems to optimize against.

Table 4 contains the expression used for the hurdle multiplier $hm(\lambda_k)$ and its parameters λ_0 and λ_1. While the hurdle multiplier is used to guarantee termination of IHHO($\mathcal{P}$), the algorithm converged to an equilibrium state before the hurdle multiplier had any effect for almost all problems as described in Section 5.5. Therefore, no effort was made to tune λ_0 and λ_1 and the initially selected values were used for all testing.

[2]This time limit is set to ensure a timely termination of testing. The optimality tolerance is increased as initial testing revealed CPLEX12 would expend a large amount of effort closing the optimality gap after finding a good, even optimal, solution. Since our implementation has not been designed for multiple threads, the single-thread mode for CPLEX12 is used for comparability.

5.3 | Problem generator

To explore the effects of underlying network topology and other problem characteristics, an ODIMCF problem generator, ODGEN,[3] was developed. ODGEN accepts as input parameters the number of commodities, number of nodes, number of directed arcs, arc cost range (given as a minimum and 90th percentile value), and commodity demand range (given as a minimum and maximum value).

ODGEN initially assigns a random position to each node in a two-dimensional space before determining the set of arcs. Arcs are generated to form a *mesh* network topology by having the probability of an arc connecting $i \in N$ to $j \in N \setminus \{i\}$ be inversely proportional to the distance in the two-dimensional space from i to j raised to a certain power. Once set A is generated, the arc costs are set to the arc distance in the two-dimensional space scaled so that the minimum and 90th percentile arc costs match the user input. All networks are connected graphs in that a path exists from every node to every other node.[4] For every arc $a \in A$ there does not exist a path from i_a to j_a with a cost less than c_a. The networks do not contain parallel arcs.

For each commodity $k \in K$, s_k is randomly chosen from N with each node having the same probability of being chosen and t_k is similarly selected from $N \setminus \{s_k\}$. The demand d_k is chosen randomly from the specified commodity-demand range using a uniform distribution.

To ensure problem feasibility, each commodity k is routed on the shortest path from s_k to t_k based on randomly assigned arc costs that differ by commodity. The generated arc capacities are then set to the sum of the capacities required by the commodities as routed on these random-cost shortest paths.

To enable experimentation of nonuniform distribution of origins and destination, the generator allows for the specification of a percentage of vertices to be designated as hubs. A specified percentage of commodities must be routed between an origin hub and a destination hub so that the bulk of the flow will be between hubs possibly passing through other vertices in the mesh network. The remaining commodities are generated as described previously still allowing for the hub vertices to be paired with non-hub vertices. The arc capacities are determined as previously described to ensure a feasible solution exists.

5.4 | Test problem characteristics

Three test sets A, H, and L are created to evaluate the effects of $|N|$, $|A|$, and $|K|$ on IHHO($\mathcal{P}$)'s performance and to explore the method's ability to solve much larger instances than previously published. Sets A and H have the same network topologies with set A having commodity origins and destinations uniformly distributed as with a mesh network structure; set H having certain vertices acting as hubs, as found in logistics and distribution networks with higher interactions between the hub nodes. Test set L is similar to set A with uniform origin and destination distribution for commodities, but with significantly larger values for $|N|$, $|A|$, and $|K|$ to analyze IHHO($\mathcal{P}$)'s runtime performance for increasingly large problem sets. All problems are known to have feasible solutions with respect to arc capacity constraints.

The characteristics of all three test problem sets are shown in Table 5. Within sets A and H eight groups of different problems with similar characteristics (number of nodes, arcs, commodities, average commodity demand, mean arc capacity, and average node degree) are created. Within each group, five different test problems are randomly generated with identical values for $|N|$, $|A|$, and $|K|$, but with different random-number seeds. Arc costs are set with a minimum value of 10 and a value of 2000 for the 90th percentile. Commodity demands range from 5 to 25. Arc capacities are determined by ODGEN to ensure feasibility. For set H, 10 percent of vertices are hubs and 80 percent of commodities must have hub vertices as both origin and destination. The large problem set L contains six groups, also shown in Table 5. Within each group, five different test problems are randomly generated with identical values for $|N|$, $|A|$, and $|K|$ but with different random-number seeds.

5.5 | Test set results

Table 6 shows the test problem run times and final solution costs for the IHH code and CPLEX's LP relaxation and integer programming solvers for test set A. Solution times are in CPU seconds and the IHH problem times are an average of 10 runs with different random number seeds. Since IHHO($\mathcal{P}$) randomizes the commodity consideration order, 10 random-number-generator seeds are used to solve each problem instance. Each reported group's results represent 50 combinations of problem and seed.

Table 6 provides computational results for the test set A. IHHO($\mathcal{P}$) found feasible solutions to all problems and CPLEX did not find a feasible MIP solution for 11 problems and 1 LPR (linear programming relaxation) problem within the time limit. The table also provides the ratios in objective function values between IHH and the LPR and the MIP (mixed integer programming) solution values provided by CPLEX12. IHH costs averaged 3.5% higher (median 2.4%) than the noninteger LPR solutions and

[3]Source code is available from the authors upon request.
[4]For industry problems with multiple subgraphs that are not connected to each other, a solution algorithm should be applied to each such subgraph separately.

TABLE 5 Test problem sets: group characteristics

Group	$\|N\|$	$\|A\|$	$\|K\|$	0/1 variables	Constraints	$\overline{c_a}$	$\overline{d_k}$[a]	$\overline{u_a}$[b]	$\overline{degree}$	$\overline{\left(\frac{\sum_{k\in K} d_k}{\sum_{a\in A} u_a}\right)}$[c]	$\overline{\left(\frac{\overline{d_k}}{\overline{u_a}}\right)}$[d]
A1	30	90	112	10 196	3454	2331	15.18	122.5	6	0.15	0.12
A2	30	90	281	25 575	8524	2331	15.13	248.4	6	0.19	0.06
A3	30	360	1728	623 812	52 204	2372	15.09	161.6	24	0.45	0.09
A4	30	360	4320	1 559 524	129 964	2372	15.03	384.5	24	0.47	0.04
A5	120	360	257	92 781	31 204	1968	15.50	149.7	6	0.07	0.10
A6	120	360	642	231 766	77 404	1968	15.15	312.2	6	0.09	0.05
A7	120	1440	4937	7 114 221	593 884	1331	14.96	195.1?	24	0.26	0.08
A8	120	1440	12 342	17 784 826	1 482 484	1331	14.98	447.5	24	0.29	0.03
H1	30	90	112	10 196	3454	2331	15.18	149.7	6	0.64	0.10
H2	30	90	281	25 575	8524	2331	15.13	283.4	6	0.74	0.05
H3	30	360	1728	623 812	52 204	2372	15.09	207.2	24	0.63	0.07
H4	30	360	4320	1 559 524	129 964	2372	15.03	388.7	24	0.74	0.04
H5	120	360	257	92 781	31 204	1968	15.50	152.9	6	0.61	0.10
H6	120	360	642	231 766	77 404	1968	15.15	281.9	6	0.75	0.05
H7	120	1440	4937	7 114 221	593 884	1331	14.96	199.3	24	0.71	0.08
H8	120	1440	12 342	17 784 826	1 482 484	1331	14.98	448.3	24	0.75	0.03
L1	480	5760	15 360	88 473 600	7 378 560	1883.5	15.0	371.9	24	0.11	0.04
L2	480	5760	38 400	221 184 000	18 437 760	1883.5	15.0	913.5	24	0.11	0.02
L3	960	11 520	25 134	289 543 680	24 140 160	2159.5	15.0	366.1	24	0.09	0.04
L4	960	11 520	62 836	723 870 720	60 334 080	2159.5	15.0	895.7	24	0.09	0.02
L5	1920	23 040	42 535	980 006 400	81 690 240	3383.2	15.0	367.5	24	0.08	0.04
L6	1920	23 040	106 338	2 450 027 520	204 192 000	3383.2	15.0	902.4	24	0.08	0.02

[a]Average demand per commodity.
[b]Mean arc capacity.
[c]Average ratio of total routed demand to combined capacity of all network arcs.
[d]Average ratio of mean destination demand to mean arc capacity.

a mean of 3.1% (median 2.0%) above MIP solution values. But these high-quality IHH solutions were identified in a fraction of the time required by CPLEX12, as shown later.

Table 7 provides computational results for the hub-network test set H. Again, IHHO($\mathcal{P}$) found feasible solutions to all problems and CPLEX did not find a feasible MIP solution for ten problems within the 2-h time limit. IHH costs averaged 2.7% higher (median 1.9%) than the noninteger LPR solutions and a mean of 2.3% (median 1.4%) above MIP solution values. But these high-quality IHH solutions were quickly identified.

The solution times for test sets A and H are summarized in Table 8, where the IHH code's best, mean, and worst running times by problem group are shown in columns 3–5. The average ratios of IHHO($\mathcal{P}$)'s running time to LPR and MIP running times are shown in the last two columns (where feasible LPR or MIP solutions exist). For set A, the ratios indicate that the average IHH solution time is 172 times faster than the CPLEX LPR solver and 565 times faster than the CPLEX MIP code, which could not find an integer solution for 20% of the problems. The longest IHH solve time for any combination of problem and seed is 52.58 s for a problem with 17 772 480 binary decision variables, 12 342 commodities, and 1440 arcs.

The hub test set H gave similar results, with IHH solving these problems 129 times faster than LPR and 506 times quicker than the MIP solvers. Although the problem dimensions are the same as those for test set A, these seem to be easier problems for IHH, hence smaller solve times.

5.6 | Statistical analyses

Statistical analysis of results is performed using SAS Version 9.4 to test whether IHHO($\mathcal{P}$)'s performance with respect to time and solution quality is significantly different for the various problem groups. As solutions are not available from LPR and MIP for all problems, the least-squares GLM procedure is used to analyze such unbalanced data. The analysis reveals whether the differences between the observed means of populations, the groups, are statistically significant. Using Tukey's significant difference test, each population is given a letter representing its ranking. Populations with the same letter do not have statistically significant differences between their means. More than one letter indicates a population's mean is not significantly different than the means of more than one distinct set of populations. Members labeled "A" have the best values, lower objective function or running times. Values become progressively worse in alphabetical order.

TABLE 6 Test set A problems' solution times and costs

		Solution times (s)			Solution cost			IHH ratio[b] to:	
Grp	**Prob**	**IHH[a]**	**LPR**	**MIP**	**IHH[a]**	**LPR**	**MIP**	**LPR**	**MIP**
A1	A1	0	0.09	1.27	25 888 055	25 219 739	25 380 948	1.026	1.020
	A2	0	0.04	0.45	13 767 055	13 619 834	13 683 181	1.011	1.006
	A3	0	0.06	0.36	15 409 031	14 307 332	14 433 630	1.077	1.068
	A4	0.1	0.11	5.47	17 565 724	16 979 419	17 091 778	1.035	1.028
	A5	0	0.06	1.41	5 999 320	5 829 935	5 851 626	1.029	1.025
A2	A11	0.01	0.27	4.12	65 433 671	64 183 982	64 382 083	1.019	1.016
	A12	0.01	30.68	0.81	32 559 908	32 134 593	32 210 139	1.013	1.011
	A13	0.01	0.14	0.88	35 452 909	34 619 069	34 723 468	1.024	1.021
	A14	0.02	0.1	0.93	45 972 645	45 424 869	45 498 683	1.012	1.010
	A15	0.02	0.09	3.34	14 470 213	14 211 330	14 238 794	1.018	1.016
A3	A21	0.9	29.8	931.37	139 968 253	131 959 935	132 849 007	1.061	1.054
	A22	0.68	12.65	839.59	64 932 431	61 226 836	61 846 681	1.061	1.050
	A23	0.71	4.97	329.55	78 437 815	74 377 380	74 804 983	1.055	1.049
	A24	0.75	10.6	854.71	95 362 616	90 247 267	90 659 190	1.057	1.052
	A25	0.66	17.34	2144.00	107 652 301	100 924 979	101 866 634	1.067	1.057
A4	A31	2.01	153.18	1110.44	328 010 021	320 334 827	[c]	1.024	[c]
	A32	1.81	91.07	1110.44	154 170 274	150 964 798	151 805 158	1.021	1.016
	A33	1.73	86.92	885.83	186 592 310	183 373 456	184 218 518	1.018	1.013
	A34	1.71	53.79	6578.58	228 166 279	223 715 157	225 323 285	1.020	1.013
	A35	2.21	162.12	1493.71	250 631 458	245 005 840	246 759 411	1.023	1.016
A5	A41	0.07	2.4	11.37	75 910 921	70 833 506	71 216 145	1.072	1.066
	A42	0.06	0.78	4.04	75 772 046	72 847 682	73 275 906	1.040	1.034
	A43	0.05	0.44	5.48	52 370 346	49 806 807	50 229 531	1.051	1.043
	A44	0.06	1.01	5.34	73 467 522	68 774 329	69 031 890	1.068	1.064
	A45	0.06	0.95	79.36	58 605 039	53 891 427	54 589 672	1.087	1.074
A6	A51	0.17	15.73	68.43	174 513 790	170 467 198	171 025 999	1.024	1.020
	A52	0.19	4.18	41.6	179 542 511	177 252 846	177 679 960	1.013	1.010
	A53	0.16	2.66	10.04	119 258 159	118 218 052	118 436 013	1.009	1.007
	A54	0.18	7.34	36.53	171 600 499	168 819 235	169 346 548	1.016	1.013
	A55	0.21	4.76	72.07	130 883 159	128 197 322	128 819 852	1.021	1.016
A7	A61	10.25	4073.79	7201.14	163 207 827	156 309 412	[c]	1.044	[c]
	A62	10.71	1894.73	7200.50	300 256 560	290 068 077	[c]	1.035	[c]
	A63	9.8	1881.49	7200.91	338 067 201	323 369 229	[c]	1.045	[c]
	A64	11.58	2211.85	7201.62	197 008 935	187 343 861	[c]	1.052	[c]
	A65	12.96	4731.51	7471.79	255 714 681	241 009 345	[c]	1.061	[c]
A8	A71	32.01	[c]	[c]	391 461 947	[c]	[c]	[c]	[c]
	A72	31.09	12 008.82	7201.61	716 857 110	707 549 277	[c]	1.013	[c]
	A73	34.43	11 991.83	7201.44	810 760 974	797 302 300	[c]	1.017	[c]
	A74	32.31	22 433.22	7202.14	474 663 944	466 662 957	[c]	1.017	[c]
	A75	34.41	19 625.22	7205.37	617 667 994	604 037 347	[c]	1.023	[c]

[a]Mean of 10 IHH runs with different random number seeds.
[b]Ratio of mean IHH minimum cost to LPR and MIP solution values.
[c]No feasible solution found by CPLEX within 2-h time limit.

Table 8's column 5 gives the ranking of problem groups within each test set based on solution time. The most difficult set A problem groups were A7 and A8, denoted by their C and D rankings. Similarly, for set H, groups H7 and H8 had the longest solution times, however the times for groups H1–H6 were not significantly different.

Table 9 shows the ratio of the IHHO($\mathcal{P}$) and LPR and MIP objective function values. These ratios indicate the percentage difference between the IHH solution value and the corresponding value of the linear programming relaxation solution or the MIP results. For example, a ratio of 1.036 shows that the IHH solution cost averaged 3.6% larger than that of CPLEX12's linear programming relaxation or its mixed integer programming solution value. For test set A, IHHO($\mathcal{P}$)'s costs averaged 3.5% higher than the LPR solution value but, as noted above, this was found 172 times faster. Similarly, the IHH costs averaged 3.1% higher

TABLE 7 Test set H problems' solution times and costs

		Solution times (s)			Solution cost			IHH ratio[b] to:	
Grp	**Prob**	**IHH[a]**	**LPR**	**MIP**	**IHH[a]**	**LPR**	**MIP**	**LPR**	**MIP**
H1	H1	0	0.18	29.42	29 955 889	28 539 832	28 618 951	1.050	1.047
	H2	0	0.12	2.42	18 732 629	17 964 677	18 024 340	1.043	1.039
	H3	0	0.04	0.57	22 622 073	22 422 453	22 447 019	1.009	1.008
	H4	0	0.05	4.3	22 378 956	21 344 286	21 378 337	1.048	1.047
	H5	0	0.11	2.32	6 558 798	5 935 733	5 960 416	1.105	1.100
H2	H11	0.01	0.19	5.58	70 538 736	68 777 857	68 980 048	1.026	1.023
	H12	0.01	0.14	2.15	44 655 527	44 492 326	44 504 599	1.004	1.003
	H13	0.01	0.15	0.68	54 413 060	54 189 697	54 260 908	1.004	1.003
	H14	0.01	0.15	0.82	50 389 822	50 067 234	50 157 382	1.006	1.005
	H15	0.01	0.11	2.41	13 944 509	13 726 647	13 764 490	1.016	1.013
H3	H21	0.36	9.43	159.31	144 381 818	142 649 463	143 598 565	1.012	1.005
	H22	0.3	9.85	145.67	65 016 267	64 098 126	64 593 555	1.014	1.007
	H23	0.21	3.54	57.74	80 029 905	79 211 507	79 348 400	1.010	1.009
	H24	0.32	8.65	632.74	102 986 334	102 197 640	102 316 526	1.008	1.007
	H25	0.39	9.79	148.4	104 165 646	102 197 678	102 722 378	1.019	1.014
H4	H31	0.85	30.96	922.14	327 687 154	325 837 074	326 010 526	1.006	1.005
	H32	0.86	51.32	310.74	133 735 869	132 775 058	133 324 942	1.007	1.003
	H33	0.79	12.2	47.12	191 168 482	190 474 085	190 602 451	1.004	1.003
	H34	0.65	17.5	62.38	238 725 874	237 934 570	238 072 969	1.003	1.003
	H35	0.57	21.39	72.11	233 146 966	232 279 190	232 412 534	1.004	1.003
H5	H41	0.05	1.87	15.51	69 673 389	65 287 854	65 742 141	1.067	1.060
	H42	0.04	1.8	23.81	76 545 065	73 030 726	73 457 452	1.048	1.042
	H43	0.05	1.61	37.88	60 697 045	57 730 345	58 155 068	1.051	1.044
	H44	0.05	2.29	36.22	80 271 608	75 536 053	75 863 431	1.063	1.058
	H45	0.04	2.53	39.21	62 502 443	59 563 649	59 901 249	1.049	1.043
H6	H51	0.17	10.55	183.8	156 275 156	150 945 216	151 690 295	1.035	1.030
	H52	0.16	6.06	48.15	172 882 442	169 114 550	170 014 917	1.022	1.017
	H53	0.2	6.13	72.72	141 093 921	137 522 791	138 217 544	1.026	1.021
	H54	0.17	6.15	41.71	190 909 105	187 478 359	188 135 845	1.018	1.015
	H55	0.16	9.08	44.59	153 301 857	149 465 335	150 208 616	1.026	1.021
H7	H61	8.95	1202.56	7201.74	161 927 789	153 606 071	[c]	1.054	[c]
	H62	7.66	978.74	7489.69	315 966 130	308 178 610	[c]	1.025	[c]
	H63	7.94	1249.44	7201.77	355 177 320	343 644 420	[c]	1.034	[c]
	H64	9.42	1766.00	7200.75	208 215 692	198 542 839	[c]	1.049	[c]
	H65	8.24	2285.61	7200.69	275 247 104	264 016 690	[c]	1.043	[c]
H8	H71	26.36	9806.65	7202.07	391 242 593	383 969 884	[c]	1.019	[c]
	H72	23.22	10 432.18	7202.02	794 124 185	783 184 975	[c]	1.014	[c]
	H73	24.48	13 263.41	7202.30	881 780 074	868 298 779	[c]	1.016	[c]
	H74	28.2	19 977.42	7202.61	521 002 962	511 999 669	[c]	1.018	[c]
	H75	24.76	18 301.86	7202.07	691 022 207	678 169 628	[c]	1.019	[c]

[a]Mean of ten IHH runs with different random number seeds.
[b]Ratio of mean IHH minimum cost to LPR and MIP solution values.
[c]No feasible solution found by CPLEX within 2-h time limit.

than the MIP values, but were found 565 times quicker, per Table 8. The Tukey rankings in column 5 do not reveal an obvious pattern as to what might make some problems more difficult.

To further explore the problem characteristics that affect IHHO($\mathcal{P}$) solution times, a regression analysis was performed based on data from test sets A and H. The observed solution time was the dependent variable and the explanatory variables were those from Table 5: $|N|$, $|A|$, $|K|$, number of binary variables and constraints, average cost, average demand, average arc capacity, and the network topology (hub or non-hub). The regression has an $r^2 = 0.9795$. Only three of the nine explanatory variables are not statistically significant: capacity, cost, and degree. Of the six significant predictors, solution time increased with $|A|$ and the number of problem constraints, but decreased if a hub topology was used, or if $|N|$, $|K|$, or the number of binary variables increased.

TABLE 8 Problem group solution time: IHH mean, LPR and MIP ratios

	IHH running time (s)				LPR:IHH	MIP:IHH
Group	Mean	Best	Worst	Rank*	time ratio[a]	time ratio[b]
A1	0.02	0.00	1.00	A	3.4	85.3
A2	0.01	0.00	0.03	A	466.9	150.5
A3	0.74	0.46	1.57	A,B	20.4	1377.4
A4	1.89	1.25	3.08	B	60.2	1171.3
A5	0.06	0.04	0.11	A	19.0	360.4
A6	0.18	0.11	0.37	A	37.7	248.8
A7	11.06	6.66	20.76	C	267.5	[c]
A8	32.85	22.24	52.58	D	502.7	[c]
Average					172.2	565.6
H1	0.00	0.00	0.01	A	∞	∞
H2	0.01	0.00	0.03	A	14.8	232.8
H3	0.32	0.17	0.77	A	25.8	715.0
H4	0.74	0.48	1.97	A	36.0	382.3
H5	0.05	0.03	0.07	A	40.4	610.5
H6	0.17	0.12	0.26	A	44.7	460.0
H7	8.44	6.37	13.61	B	177.3	860.1
H8	25.40	18.77	38.58	C	565.2	283.6
Average					129.2	506.3

[a](Mean LPR solution time)/(mean IHH solution time).
[b](Mean MIP solution time)/(mean IHH solution time).
[c]No feasible solution found by CPLEX12 within 7200 s.
*Tukey's significant difference test ranking.

TABLE 9 IHH solution value ratio to LPR, MIP

	IHH:LPR cost ratio[a]				IHH:MIP cost ratio[b]		
Group	Mean	Best	Worst	Rank	Mean	Best	Worst
A1	1.036	1.027	1.048	A,B	1.029	1.021	1.041
A2	1.017	1.012	1.024	A,B	1.015	1.010	1.021
A3	1.060	1.055	1.067	B	1.052	1.047	1.059
A4	1.021	1.020	1.023	A,B	[c]	[c]	[c]
A5	1.064	1.047	1.080	B	1.056	1.040	1.072
A6	1.017	1.013	1.020	A	1.013	1.010	1.017
A7	1.047	1.046	1.050	A,B	[c]	[c]	[c]
A8	1.017	1.017	1.018	A,B	[c]	[c]	[c]
Average	1.035	1.030	1.042		1.031	1.024	1.038
H1	1.051	1.039	1.064	C,B	1.048	1.036	1.061
H2	1.011	1.009	1.014	A	1.009	1.007	1.012
H3	1.013	1.011	1.015	A	1.008	1.006	1.010
H4	1.005	1.004	1.005	A	1.003	1.003	1.004
H5	1.056	1.044	1.066	C	1.049	1.037	1.060
H6	1.026	1.019	1.034	B,A	1.021	1.014	1.029
H7	1.041	1.039	1.044		[c]	[c]	[c]
H8	1.017	1.016	1.018		[c]	[c]	[c]
Average	1.027	1.022	1.033		1.023	1.017	1.029

[a](IHH objective function value)/(LPR Objective function value).
[b](IHH objective function value)/(MIP objective function value).
[c]No feasible solution found by CPLEX12 within 7200 s.

TABLE 10 Large problem set L: solution time, cost, coefficient of variation

Group	Problem	IHH time	IHH cost	CV cost
L1	L51	77.78	1 426 590 347	0.00018
	L52	81.73	1 254 903 691	0.00013
	L53	73.63	1 703 192 064	0.00014
	L54	81.48	1 955 056 175	0.00017
	L55	72.03	1 434 497 785	0.00013
	Average:	77.33		0.00015
L2	L56	226.81	3 562 506 827	0.00002
	L57	213.14	3 123 060 684	0.00005
	L58	218.98	4 236 881 946	0.00004
	L59	205.03	4 861 652 396	0.00004
	L60	208.00	3 552 577 022	0.00004
	Average:	214.39		0.00004
L3	L71	286.58	3 660 398 107	0.00012
	L72	294.18	3 793 204 910	0.00011
	L73	287.23	2 713 212 910	0.00013
	L74	275.33	4 116 477 552	0.00008
	L75	284.63	3 766 807 718	0.00010
	Average:	285.59		0.00011
L4	L76	820.17	9 119 420 821	0.00003
	L77	764.44	9 397 310 591	0.00002
	L78	813.81	6 720 114 699	0.00002
	L79	820.91	10 235 141 998	0.00004
	L80	783.75	9 318 572 755	0.00003
	Average:	800.62		0.00003
L5	L91	1114.24	6 497 960 761	0.00008
	L92	1222.81	20 287 326 224	0.00007
	L93	1132.24	8 091 918 047	0.00007
	L94	1141.31	15 842 224 803	0.00009
	L95	1156.20	8 657 046 549	0.00006
	Average:	1153.36		0.00007
L6	L96	3118.31	16 154 800 326	0.00003
	L97	3210.38	50 496 932 124	0.00003
	L98	3306.78	20 132 144 074	0.00002
	L99	3141.18	39 421 362 436	0.00002
	L100	3181.98	21 458 101 592	0.00003
	Average:	3191.72		0.00002

Additional analysis was performed on results of test set A to determine the impact of the $hm(\lambda_k)$ function on overall performance. As noted in Sections 4.3.1 and 4.5, $hm(\lambda_k)$ is designed to force IHHO($\mathcal{P}$) to converge to an equilibrium if the market forces are insufficient. The code was modified to count the number of times the hurdle multiplier prevented a commodity from switching to a new route so when $rc(CP(k), k, A, \mathbf{X}) \geq hm(\lambda_k) rc(CP(k), k, A, \mathbf{X})$ while $rc(NewPath, k, A, \mathbf{X}) < rc(CP(k), k, A, \mathbf{X})$. For the 400 runs of IHHO($\mathcal{P}$) performed, the hurdle multiplier had an effect during five calls to Route(k, λ_k, $\mathcal{P}$). These five instances occurred for one seed of one problem in Group A8. These results indicate that for most problems and commodities IHH's price mechanism alone is sufficient to reach an equilibrium.

Analysis was also performed to determine the impact of the ResidualCapacityClearance($\mathcal{P}$) procedure on overall performance. As noted in Section 4.3.2, the marginal market cost curve is not an exact match to the original ODIMCF problem and ResidualCapacityClearance($\mathcal{P}$) attempts to close the gap by finding lower original cost paths to fully utilize arc capacities and by rerouting commodities that paid a large marginal market cost to exceed an arc's capacity constraint. For all runs of problem set A, 69.25% of solutions after IHHO($\mathcal{P}$)'s main loop are feasible with respect to arc capacity constraints. For the infeasible solutions, the mean and median percentage of arc capacity constraints violated are 0.75% and 0.56% respectively with ResidualCapacityClearance($\mathcal{P}$) able to resolve all violations. For the feasible solutions, the mean and median percentage of ODIMCF objective function improvement after ResidualCapacityClearance($\mathcal{P}$) are 0.66% and 0.70% respectively.

For all runs, the mean and median number of reroutes per commodity during ResidualCapacityClearance(P) are 0.090 and 0.089 respectively indicating most commodities do not change routes during ResidualCapacityClearance($\mathcal{P}$). These results indicate that the main loop of the IHHO($\mathcal{P}$) algorithm based on the invisible-hand analogy is doing most of the work with ResidualCapacityClearance($\mathcal{P}$) closing the small remaining gap.

5.7 | Large problems: test set L

Test set L contains the largest ODIMCF problems solved to explore the capabilities of the IHH algorithm. Because these problems are beyond the solvability of CPLEX, they are only run using IHHO($\mathcal{P}$). The problems dimensions are given in Table 5 with Table 10 providing solution times, costs, and coefficients of variation from the computer experiments. These contain the largest ODIMCF problem instances published to date, with group L6 networks containing 1920 nodes, 23 040 arcs, and 106 338 commodities, as can be found in industrial problems [35].

The results show that the IHH can even solve problems with over 2 billion binary variables and 200 million constraints in 3200 s. If the results from test sets A and H continue to hold, the solution values could be within a few percentage points of the exact solution values.

To assess the impact of the inherent randomness in IHH, each problem was solved 10 times with different random number seed values and the average time and cost reported. An evaluation of the computational results of multiple runs per problem shows that the mean coefficient of variation CV (standard deviation normalized by the mean) for test sets A and H of IHH solution values is 0.35% (median is 0.00201), reflecting small variation in the resulting solution values; for the largest set L, the even smaller average CV = 0.00007, shown in Table 10. This lack of variation indicates that the IHH provides robust results for these problems.

6 | CONCLUSIONS

Origin-destination integer multicommodity flow problems occur in a variety of application areas—including logistics and telecommunications—and are often of large dimensions, in terms of nodes, arcs, commodities, binary variables, and constraints. This heuristic, inspired by Adam Smith's invisible hand description of the efficient economic processes underlying a market economy, provides a new method for solving large-scale instances of such problems. Computational testing demonstrates it has the capability to achieve integer feasible solutions with excellent solution quality relative to objective function value.

The current implementation's time performance appears competitive over a wide range of problem characteristics. Linear space requirements combined with fast running times enables IHH to solve realistic problems with millions of constraints and hundreds of millions of binary variables that are beyond the reach of other methods. The methodology is shown to identify high-quality integer solutions quickly, as verified in comparisons with state-of-the-art commercial software.

Further research into this approach could take advantage of parallel computing implementations whereby a host of competing system processes could emulate the distributed decision-making of a marketplace. By replacing randomized choices with algorithmic race conditions, the method might converge faster while still uncovering high-quality solutions and enable the solution of even larger problem instances. Further research could examine procedures to dynamically alter the marginal market cost curves for each arc during algorithm execution reacting to actual demand and fully exploiting the rationing function of the demand curve.

Additional research directions could evaluate application of this approach to variations of the ODIMCF problem itself. One variation would involve problems where the underlying network does not have sufficient capacity to accommodate all commodities and a decision on which commodities to service must be made. Such work could include reporting on the state of the problem to inform decisions on future changes to the network. Other versions to study would include problems where there is a limit on the total route length for commodities and those with an imposed commodity demand schedule so that demand may vary over time, possibly with commodities completely deactivating during certain periods.

ORCID

Richard S. Barr https://orcid.org/0000-0002-1925-6642
Thomas McLoud https://orcid.org/0000-0003-3706-2179

REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Some recent advances in network flows*, SIAM Rev. **33** (1991), 175–219.

[2] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows: Theory, algorithms, and applications*, Prentice-Hall, Upper Saddle River, NJ, 1993.

[3] A. Amiri and R. Barkhi, *The multi-hour bandwidth packing problem*, Comput. Oper. Res. **27** (2000), 1–14.

[4] A. Amiri, E. Rolland, and R. Barkhi, *Bandwidth packing with queuing delay costs: Bounding and heuristic solution procedures*, Eur. J. Oper. Res. **112** (1999), 635–645.

[5] A. Assad, *Multicommodity network flows: A survey*, Networks **8** (1978), 37–91.

[6] C. Barnhart, C. A. Hane, and P. H. Vance, *Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems*, Oper. Res. **48** (2000), 318–326.

[7] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear programming and network flows*, 4th ed., John Wiley & Sons, Hoboken, NJ, 2010.

[8] E. V. Bowden, *Economics: The science of common sense*, South-Western Publishing, Cincinnati, OH, 1989.

[9] O. Brun, B. Prabhu, and J. Vallet, *A penalized best-response algorithm for nonlinear single-path routing problems*, Networks **69** (2017), 52–66.

[10] D. S. Chen, R. G. Batson, and Y. Dang, Eds., *Applied integer programming: Modeling and solution*, John Wiley & Sons, Hoboken, NJ, 2010.

[11] D. Corne, M. Dorigo, and F. Glover, *New ideas in optimization*, The McGraw-Hill Companies, St. Louis, 1999.

[12] B. Fortz, E. Gorgone, and D. Papadimitriou, *A Lagrangian heuristic algorithm for the time-dependent combined network design and routing problem*, Networks **69** (2017), 110–123.

[13] R. Garlick and R. Barr, *Dynamic wavelength routing in WDM networks via ant colony optimization*, in *Ant algorithms*, M. Dorigo, G. D. Caro, and M. Samples, Eds., Springer-Verlag, Berlin, 2002, 250–256.

[14] M. Gendreau and J. Y. Potvin, Eds., *Handbook of metaheuristics*, Springer, New York, 2010.

[15] M. K. Girish, B. Zhou, and J. Q. Hu, *Formulations of the traffic engineering problems in MPLS based IP networks*, Proceedings of the Fifth IEEE Symposium on Computers and Communications, 2000, pp. 214–219.

[16] C. L. Huntley, D. E. Brown, D. E. Sappington, and B. P. Markowicz, *Freight routing and scheduling at CSX transportation*, Interfaces **25** (1995), 58–71.

[17] J. L. Kennington and R. V. Helgason, *Algorithms for network programming*, John Wiley & Sons, New York, 1980.

[18] E. Kieffer, G. Danoy, P. Bouvry, and A. Nagih, *A new modeling approach for the biobjective exact optimization of satellite payload configuration*, Int. Trans. Oper. Res. **26** (2019), 180–199.

[19] M. Laguna and F. Glover, *Bandwidth packing: A tabu search approach*, Manag. Sci. **39** (1993), 492–500.

[20] T. Li, *MPLS and the evolving internet architecture*, IEEE Commun. Mag. **37** (1999), 38–41.

[21] N. Mankiw, *Principles of economics*, 9th ed., Cengage, Boston, 2021.

[22] E. Morenoa, A. Beghelli, and F. Cugini, *Traffic engineering in segment routing networks*, Comput. Netw. **114** (2017), 23–31.

[23] L. F. Moura, L. P. Gaspary, and L. S. Buriol, *A branch-and-price algorithm for the single-path virtual network embedding problem*, Networks **71** (2018), 188–208.

[24] G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization*, Wiley, New York, 1988.

[25] K. Park, S. Kang, and S. Park, *An integer programming approach to the bandwidth packing problem*, Manag. Sci. **42** (1996), 1277–1291.

[26] S. Raghavan and D. Stanojević, *Branch and price for WDM optical networks with no bifurcation of flow*, INFORMS J. Comput. **23** (2011), 56–74.

[27] M. Resende and C. C. Ribeiro, Eds., *Optimization by GRASP: Greedy randomized adaptive search procedures*, Springer-Verlag, New York, 2016.

[28] M. G. C. Resende and C. C. Ribeiro, *A GRASP with path-relinking for private virtual circuit routing*, Networks **41** (2003), 104–114.

[29] E. Rolland, A. Amiri, and R. Barkhi, *Queueing delay guarantees in bandwidth packing*, Comput. Oper. Res. **26** (1999), 921–935.

[30] H. Saito, Y. Miyao, and M. Yochida, *Traffic engineering using multiple multipoint-to-point LSPs*, IEEE INFOCOM **41** (2000), 894–901.

[31] A. Smith, in *An inquiry into the nature and causes of the wealth of nations*, W. Strahan and T. Cadell, London, 1776.

[32] C. Snyder and W. Nicholson, *Microeconomic theory: Basic principles and extensions*, 10th ed., South-Western Publishing, Mason, OH, 2008.

[33] D. P. Williamson, *Network flow algorithms*, Cambridge University Press, Cambridge, UK, 2019.

[34] L. Wolsey, *Integer programming*, John Wiley & Sons, New York, 1998.

[35] S. Yasukawa, A. Farrel, and O. Komolafe, RFC 5439: An analysis of scaling issues in MPLS-TE core networks, Request for comments, Network Working Group, https://tools.ietf.org/html/rfc5439, 2009.

How to cite this article: Barr RS, McLoud T. The invisible-hand heuristic for origin-destination integer multicommodity network flows. *Networks.* 2021;1–19. https://doi.org/10.1002/net.22026