

Analyses of Advanced Iterated Tour Partitioning Heuristics for Generalized Vehicle Routing Problems

Anupam Seth[†], Diego Klabjan^{*}, Placid M. Ferreira[†]

[†] Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, IL

^{*} Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL

June 26, 2011

Abstract: Theoretical analyses of a set of iterated-tour partitioning vehicle routing algorithms applicable to a wide variety of commonly-used vehicle routing problem variants are presented. We analyze the worst-case performance of the algorithms and establish tightness of the derived bounds. Among other variants we capture the cases of pick-up and delivery, and multiple depots. We also introduce brand new concepts such as mobile depots, partitioning of customer nodes into groups, and potential opportunistic under-utilization of vehicle capacity by only partially loading the vehicle, among others, which arise from a printed circuit board application. The problems studied are of critical importance in many practical applications.

Keywords: vehicle routing, worst-case analysis, printed circuit board, approximation algorithms

1 Introduction

Logistics in today's increasingly global economies has become a more important field than ever before. Even though logistics in itself has a very significant business impact, nevertheless, the breadth of applications of vehicle routing goes well beyond the traditional trucking industry. Vehicle routing algorithms play an important role in vehicle routing related information systems. Theoretical analyses of vehicle routing problems have led to significant insights into the nature and complexity of the proposed algorithms and they help in better understanding the potentials and limitations of such algorithms.

A novel and recent area of applications of vehicle routing is production planning of printed circuit board assembly on an automated (primarily surface-mount type) equipment, for example, on the collect-and-place or pick-and-place type machines [27]. More specifically, collect-and-place machines are typically used for high-volume, high-flexibility production and are becoming increasingly popular in industrial settings because of their inherently advantageous characteristics and capabilities. The placement sequencing problem, i.e., the problem of planning the pick-up and delivery of electronic components or chips from feeder trays on the side of the machine onto a bare printed circuit board (PCB) in the center on such machines is an NP-hard problem that turns out to be essentially a generalization of the vehicle routing problem (VRP) in several different dimensions. Several interesting variants appear based on the manufacturing scenario and configuration of the machine, which are generalizations of many well-known and standard vehicle routing problem variants that have been studied in the past. Thus, existing algorithms and research do not directly apply to the problems motivated by the PCB context, but they are interesting generalizations of many important standard VRP variants, and hence a study of these problems is of great value.

PCB manufacturing plays a very important role in today's economy. Global revenues for the PCB industry exceeded \$50 billion in 2007 and are expected to reach more than \$76 billion in 2012 [22]. A PCB is a flat board that carries the chips and various other electronic components. This board is made up of alternating layers of copper and plastic, with the etching process performed on the copper layers to provide interconnects. These boards are capable of holding several components depending on the required specifications and produce complex interconnections. PCBs can be of different sizes and varying densities and are manufactured in automated assembly lines where high-speed placement machines put components

on the boards. A line can assemble components on multiple types of PCBs and it has one or several high speed machines to perform the actual placement of operations [26]. The assembly of PCBs is a complex process involving the placement of hundreds (even up to a thousand or more) of electronic components in different shapes and sizes at specific locations on the board. These assembly lines, and more specifically, collect-and-place machines (see Figure 1) form the bottleneck in production, thus even small savings in total assembly time can significantly improve the bottom-line.

These machines have one or more rotary heads (see Figure 3) that work in tandem to pick up components or chips from a feeder tray (or an array of feeder slots each feeding one particular type of component) and deliver them to specific locations on the bare circuit board as dictated by the specific design requirements (illustrated in Figure 2). Each rotary head contains typically between 5 to 20 spindles, or placeholders for the chips, thus limiting the capacity of the “vehicle”. There are additional complicating manufacturing constraints and specifications that make the problem much more complex and harder to solve than just a basic vehicle routing problem.

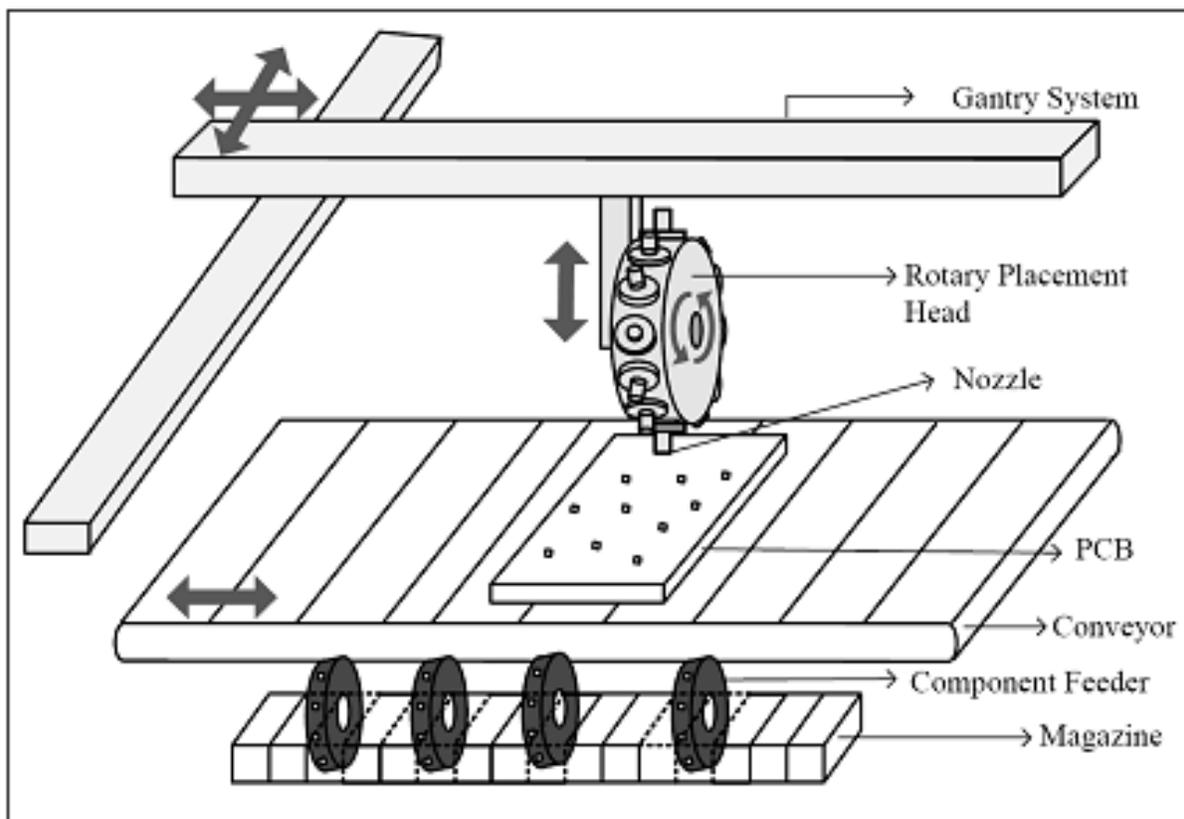


Figure 1: A collect-and-place machine

In this paper we study the placement sequencing problem on a single machine processing a single board at a time with a single machine head. We assume that the arrangement of components or chips on the feeder magazine is given, i.e., the supply characteristics of each depot are fixed beforehand, which is an assumption frequently used in the PCB literature. Under these specific constraints posed by the scoped configuration of the machine, the placement sequencing problem is very similar to a complex, yet striking version of the vehicle routing problem with a single capacitated vehicle (in our case, the rotational robotic head with multiple spindles loaded onto a dual gantry) performing multiple trips from the depot (in our case, the feeder magazine) to satisfy customer demand (in our case, the design requirements for the board being populated, stipulating the specific component type to be inserted at each placement location).

As already alluded to, the problem in its general form in the given setting captures many important VRP

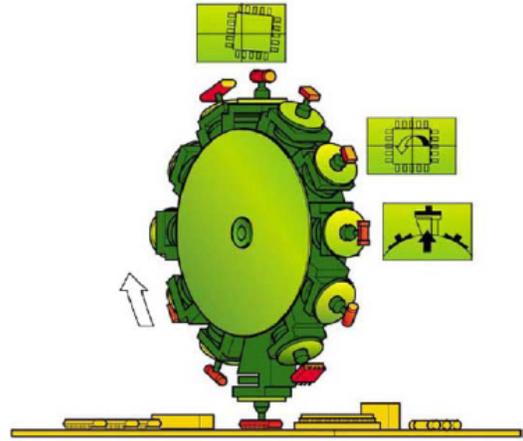
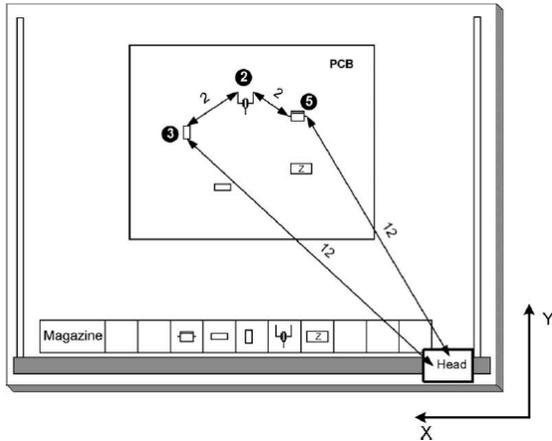


Figure 2: A schematic of the collect-and-place machine [13] Figure 3: Detailed view of a rotary collect-and-place head [28]

variants. The first major difference from all standard VRP variants is the introduction of a moving depot that utilizes the time that vehicle is away to reposition itself to a more advantageous location. Thus, classical VRP is a special case when the depot movement velocity is fixed to zero. The second major extension lies in the generalization of the underlying network to use groups of customer locations, only one location in each pre-defined group needs to be visited by the vehicle. We take this generalization one step further and consider also groups of depot locations such that a vehicle visits only one location in a group. Such problems have been commonly studied in the context of the traveling salesman problem (TSP) where they are known as generalized TSP (GTSP) [14], [11], but not so much in the VRP context even though they have several important practical applications. Apart from these differences, in the modeled problem, if we consider only one possible pick-up and delivery point for each type of a component, the problem essentially becomes a static dial-a-ride problem in which all requests for transportation are known at the beginning. This scenario can also be analyzed as a vehicle routing problem with pick-up and delivery since there are no extra side-constraints typically found in the dial-a-ride problem. To this scenario, if we further add the restriction that all pick-ups precede all drop-offs in a route, it can be seen that the problem reduces to a vehicle routing problem with backhauls type scenario. If we further simplify the problem and assume only one component type, i.e., a single commodity problem, we get the standard multi-depot vehicle routing problem. Finally, if we assume a single pick-up point for all components in addition to the previous simplifications, we end up with the standard capacitated VRP.

Other interesting VRP variants that have not been studied in literature ever before also arise as special cases, for instance, the VRP with moving depots and the generalized VRP (GVRP), which is an extension of the VRP with an underlying GTSP replacing the standard TSP, potentially both on the pick-up and delivery side. A GVRP is a VRP where the underlying network is a GTSP, i.e., there are clusters of customer and depot locations, and the vehicle needs to only visit one location in each cluster. We stress that on the customer side, every cluster would need to be visited exactly once, but such a restriction is absent on the depot side.

We next formally state GVRP where it is assumed that each customer and depot is associated with a unique location. We are given clusters $C_i, i = 1, \dots, n$ of customers, with cluster C_i having n_i customers, and each customer exhibiting a unit demand. We also have clusters $D_j, j = 1, \dots, m$ of depots, each cluster D_j having m_j depot locations and each individual depot storing a single unit of demand. A vehicle has capacity of q . Before starting a route to deliver goods to customers, the vehicle must pick the items to be delivered along the route by visiting depots within a single cluster D_j for a j in $1, \dots, m$. Since each depot holds a single unit of demand, it must be the case that $|D_j| \leq q$. GVRP is the problem of finding the minimum cost route through all the customer clusters and (some) depot clusters by visiting exactly one

location in each cluster of customers. After the items are delivered along a route, the vehicle must return to a cluster of depots in order to pick up the items for the next route. Various special cases of this problem yield well studied variations to the classical VRP problem. In addition, GVRP is applicable to a variety of automation and robotics related applications and a well-designed algorithm could lead to potentially large savings in such situations.

Besides the above-mentioned importances of this problem in the PCB assembly planning context, and its value in studying and deriving bounds for important VRP variants, this problem also is of striking similarity and crucial importance to related research areas including, but not limited to, seaport container problems for ship loading and unloading [30], point-to-point motion planning of robots [19] used in automated assembly lines and for operations such as painting and welding, warehouse operations [33] utilizing automated storage and retrieval systems [25] requiring the routing of autonomous guided vehicles [4], which are also used as an integral part of flexible manufacturing systems [29], and hard disk drive data retrieval and storage optimization problems [24].

1.1 Contributions and Outline

This paper presents analyses of advanced iterated-tour partitioning algorithms for generalized vehicle routing problems, which encompass many standard versions of the VRP and extend them to a generalized setting with the ability to deal with groups or clusters of customer and depot sites and a maximum attribute arising in the objective function as a result of mobile depots.

The iterated tour partitioning heuristic (ITP) in the standard VRP setting finds a TSP tour in the network and then traverses the tour. Each time the truck following the tour delivers all loaded items, it detours from the tour by going to the close-by depot, picks up new items, and returns back to the tour at the next node on the tour. We generalize this heuristic to accommodate our settings. In the most general case, we set up an auxiliary network based on the computed TSP and the final solution consists of finding a shortest path on this auxiliary network.

The most striking and unique feature of our problem is the notion of moving depots. Another innovative feature of our model is the pick-up and delivery aspect. It turns out that this can be modeled on a network with only customer nodes without explicitly modeling depot nodes by using two arc weights (one capturing the cost of the movement of the vehicle and the other one reflecting the cost of pick-ups). The cost of a VRP solution is the combination of these two arcs on the routes. We call such a network the generalized and expanded 2-cost edge network. Two arcs per pair of nodes are sufficient if each ‘truck’ must leave fully loaded. If this condition is relaxed, we need more than two arcs between two nodes and the resulting network is termed the generalized and expanded multi-cost edge network. The reason we can deal with the depots in such an implicit manner is due to the weaker condition of not requiring to visit every depot exactly once as is the case on the delivery side.

The second major contribution is the extension of the standard vehicle routing problem into the generalized setting wherein the vehicle needs to visit only one customer in each pre-defined set of customers.

These models provide the underlying framework we use to derive a generalized ITP heuristic that covers the variants examined. We present tight approximation ratios for the ITP heuristics for

- the generalized full-load case, in which the vehicles are assumed to always run on a full-load and return empty to the depot,
- the generalized partial-load case, in which vehicles are allowed to carry only as much as they intend to deliver on a given route,
- and, the generalized partial-load case with waiting cost, in which an additional cost component is present in the objective function to capture moving depots.

In analyzing the algorithms, we use a unique lower bounding procedure based on the structure of the optimal solutions to such problems coupled with an upper bound derived from ITP. To do so in a more general context, we need to define q auxiliary networks with cost structures deduced from the original problem. The entire analysis and the heuristics are based on the multi-cost network.

Our contributions first lie in the development of the modeling framework. The novel idea of introducing multi-cost networks to model pick-up operations is an important contribution towards a unified framework.

Second, we provide approximation and tightness analyses for our ITP-heuristics for the generalization of standard VRP scenarios, specifically the vehicle routing problem with pick-up and delivery. Another important contribution is the new and novel generalization of the heuristic. We generalize the problem to solve it on groups of customer sites only one of which needs to be visited by a vehicle rather than individual customers all of which must be on the vehicle’s route. Furthermore, we allow for multiple depots (and also generalize the concept of a depot to a cluster of depots), each of which is capable of moving when not replenishing a vehicle leading to the introduction of a waiting cost. And finally, we generalize the standard VRP scenario to exploit opportunities where under-utilization of the vehicle’s capacity in the short-term leads to shorter routes. One of the major accomplishments of our work is to keep the problem general and present a unified model so that approximation ratios for well-known VRP variants (for many of which there does not exist any theoretical analysis so far) can easily be reduced from our model.

The paper is structured as follows. We first finish the introduction with a literature review. Section 2 provides a broad overview of the problem, including the variants of the problem addressed in this work, and those that can be directly reduced from them. In Section 3, we present the ITP-heuristic and develop approximation ratios for each of the above-mentioned three scenarios. Tightness of the derived ratios is also proven. Finally, in Section 4 we present a numerical study.

1.2 Literature Review

To the best of our knowledge, there is no previously published research studying and presenting a model and analysis of the generalized VRP problems addressed in this paper. A few purely computational studies (see, for example, [13], [32], [17], and [18]) have been carried out, but are exclusively empirical in nature, and do not provide a significant insight into the analysis of their underlying algorithms or theoretical implications of the underlying mechanisms used to solve the problem.

ITP algorithms have been studied and applied to many different VRP scenarios. Most of this literature exists in the theoretical computer science domain, and the algorithms are well-known for their amenability to theoretical analysis. The underlying concept and the first ITP algorithm was presented by Haimovich, Rinnooy Kan and Stougie in their path-breaking paper [15] (actually, they published an earlier version of their work in [16]). The worst-case bound developed in [15] was shown to be tight by Li and Simchi-Levi [20]. This work was extended and modified later by Altinkemer and Gavish [1] and Beasley [3], but still applicable to only the standard VRP scenario. Subsequent research has seen applications and transformations to a few different hypothetical and practical settings, however, none of these have addressed the problem in its general form as we do. For example, there is individual work on ITP heuristics for multiple depot problems [20] and for problems that have both pick-up and delivery aspects [21], but there is none that addresses both aspects.

Recently, Bompadre et. al. [5] presented lower bounds for the vehicle routing problem with and without split deliveries, improving the well known bound of Haimovich, Rinnooy Kan, and Stougie. These bounds were then utilized in a design of best-to-date approximation algorithms. Chawla [8] presented new approximation algorithms as well as hardness of approximation results for several planning and partitioning problems. She gave the first approximation algorithms for problems such as orienteering, deadlines-TSP, and time-windows-TSP, as well as results for planning in stochastic graphs (Markov decision processes). However, her work is limited to TSPs, or uncapacitated VRPs. More importantly, the moving depot and the GVRP aspects have not yet been studied.

Bramel and Simchi-Levi [7] developed an asymptotically optimal heuristic method called the location based heuristic for the VRP with time windows by transforming it into a capacitated location problem with time windows. They use Lagrangian relaxation coupled with probabilistic analysis to prove their result. Earlier, they proved a variant of the location based heuristic to be also asymptotically optimal for the standard capacitated VRP [6].

Daganzo [9] presents some analytical insight into distribution patterns for the VRP with pick-up and delivery type problems at the system design level so as to potentially provide guidelines for districting and cost estimation by characterizing broad routing strategies in terms of problem characteristics independent of specific customer locations. For the VRP with pick-up and delivery, the first probabilistic analysis of a simple algorithm was presented by Stein [31]. Later on, Psaraftis [23] showed his adaptation of the minimum spanning tree heuristic for the TSP to have a worst-case performance ratio of 3.0 while also demonstrating its superior practical performance as compared to Stein’s method. Gendreau et. al. [12] analyzed the single

unit vehicle routing problem with pick-up and delivery (or, in other words, the traveling salesman problem with pick-up and delivery) on trees and cycles showing that these problems can be solved optimally in linear time, and that a tight worst-case performance ratio of 2, 3 can be achieved on undirected, complete, and triangular networks by the tree- and cycle-based heuristic, respectively.

In summary, we find that there is no prior work that addresses a practical and unified VRP scenario where there are

- simultaneous considerations of both the pick-up and the delivery aspects in routing;
- allowing only one or more locations out of mutually exclusive and exhaustive groups of several possible customer sites to be visited, i.e. delivering to groups of customer sites (such that the underlying structure of the problem becomes a GTSP rather than a vanilla TSP)
- exploiting the availability of multiple depots,
- the depots are mobile and can reposition themselves into advantageous locations,
- allowing for vehicles to have partially-utilized capacity in favor of opportunistic pick-ups at a later stage in the route.

2 Problem Statement

In the PCB context, the problem can be stated as one of finding the least cost sequence of picking-up components from the feeder magazine and placing them on the board subject to the following conditions:

- (1) all components required by the design must be placed,
- (2) every component that is placed must have been picked up previously (and not yet placed),
- (3) every spindle may contain only one component during a ‘route’, and
- (4) at no time in a route must the capacity of the head be violated.

At the same time, we want to fully exploit opportunities offered by simultaneous motion, spindle jumping (a phenomenon caused by allowing bidirectional and multi-stepped indexing motion of the placement head), and feeder magazine movement.

This can be restated in a generic VRP context as follows. Assuming

- there are n customers (i.e., board locations in our case), and
- there are m depots (i.e., feeder slots in our case) to pick up the items from,

the problem characteristics entail the following restrictions:

1. every ‘customer’ requires only one item of a specific type, i.e., there is a single-unit demand; and
2. each depot stocks a single item type, implying that travel among depots is required as the vehicle moves from depot to depot to pick up each of the required items.

The objective is to find the least cost sequence of pick-up of items from the depots and delivering them to the ‘customers’ subject to the following conditions:

- all items required by the customers must be delivered,
- every item delivered must have been picked up previously (and not yet delivered),
- the capacity limit of the vehicle must not be violated, and
- the cost implications of the loading and unloading sequence for the vehicle to be formalized later are accounted for.

A typical route is depicted in Figure 4. Each of the customer locations, as well as depot locations, can be served by any one of the spindles on the head. The choice of the spindles for the different locations is akin to the assignment of a customer on a vehicle with multiple seats, and correspondingly different “occupancy costs” for the seats, as in a commercial airliner, for example. This can be modeled by expanding the network to accommodate groups of customer and depot locations, only one of which is finally chosen in any given route. (Each node corresponds to a seat occupied by the customer and since the customer must sit in exactly one seat, a single node in a pick-up group must be selected, similarly when a customer alights, only a single seat becomes vacant, and thus only a single node in the delivery group must be selected.) Another analogy can be found in the freight industry with the notion of load positions for cargo, e.g., cargo in an aircraft belly is stored by load positions. The problem is further complicated by the introduction of mobility for depots, allowing them to re-position themselves to a more suitable location when eventually intercepted by the vehicle during the period that the vehicle is out on its tour delivering customers. This is handled by discretizing the possible locations that the depot can move to, and correspondingly adding a third dimension to the network model and consequent extra layers of nodes in the clusters, where each of the pairs of depot location-spindle combinations is now supplemented by real-time depot position information.

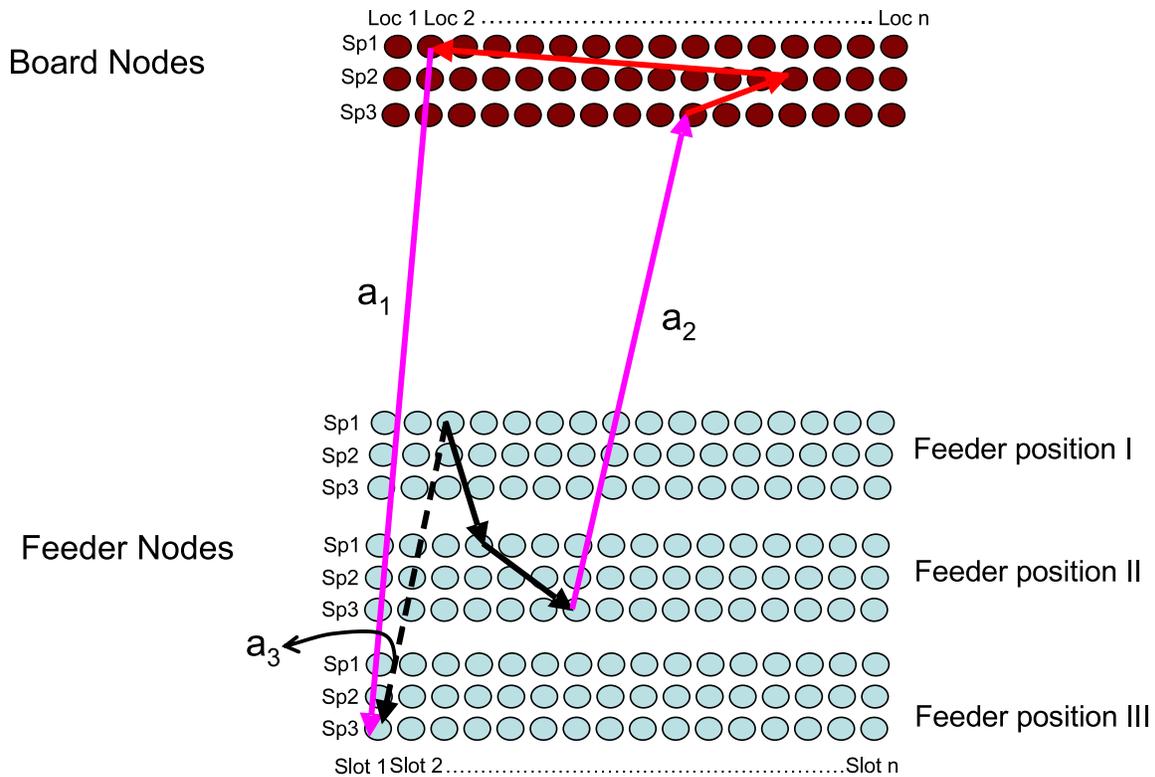


Figure 4: Example of a typical ‘route’

2.1 Network Modeling Concepts

In this section, we present the modeling concepts used to define our network models based on which the analysis has been carried out for the different cases. Based on the description aforementioned, it becomes clear that we need a generalized network model comprising of groups of nodes to represent the problem because each pick-up or drop-off location can be potentially visited by any one of the spindles on the head. Thus, each group represents a physical site or location, and the nodes within a group represent the different possibilities (of which only one can exist in a feasible route solution) of visiting that location with the spindles (based on Figure 4, we would have to have a set of board and feeder nodes). However, such a network would be thus, extremely complicated obliterating the possibility of an analytical study. Therefore, to simplify the

network model we developed an innovative idea of using a double-cost arc representation, which we later extend to a multiple-cost arc representation. This implies that there is no need for the depot nodes and their groups and thus they can be disregarding completely for the sake of the analysis. It is important to stress that this new network nicely captures the depot movements through the unconventional cost structure defined later on in this section, and does not compromise our results due to no additional or restrictive assumptions being enforced in it.

2.1.1 Formal Definition of the Generalized and Expanded 2-Cost Edge Network (*GE2-CEN*) Model

In this section, we present a generalized and expanded formulation of the two-cost edge network (*GE2-CEN*). We begin by assuming that each time the vehicle is dispatched, it must be fully loaded with exactly q items. Consider a complete undirected graph $\mathcal{G}(X, E)$, where X is the set of nodes, and E is the set of edges. Let G_1, G_2, \dots, G_K be a partition of X into K mutually exclusive and exhaustive subsets or groups. Every node $i \in X$ has a unique index $g(i) \in \{1, 2, \dots, K\}$ such that $i \in G_{g(i)}$. Each group is defined for a customer subset (board location) having multiple ‘cities’ or locations (board location and pick-up spindle combinations). We assume throughout that $n \geq K \geq q$, i.e., $\zeta = K/q \geq 1$, where q is the capacity of the vehicle. Any two nodes i and j in this network are connected by an edge with two types of costs on it - the primary cost, which is the ‘direct’ cost to go from node i to node j , and the secondary cost, which is the ‘indirect’ cost to go from node i , visiting exactly q depots en route, filling up the entire vehicle, which always is assumed to return empty, and returning to node j (see Figure 5). The direct cost $d_{i,j}$ is the ‘true’ cost of the vehicle moving from node i to node j . The indirect cost $r_{i,j}$ models the case of the truck finishing the last delivery at node i , next visiting q depots (not modeled explicitly), and then moving to node j with q items to start the next delivery. Cost $r_{i,j}$ is the cost of performing these operations in the most cost effective way.

In standard VRP settings, $d_{i,j}$ is the cost of going directly from i to j . On the other hand, $r_{i,j}$ would typically be the cost of finishing a route at node i , transitioning to the lowest cost depot and starting a new route at node j . Note that there are groups only corresponding to the customers, and thus, the depots are not represented either by nodes or groups. We define for each edge $(i, j) \in E$, the ‘direct’ cost $d_{i,j} = d_{j,i}$ between nodes i and j , and the ‘indirect’ cost $r_{i,j} = r_{j,i}$ between nodes i and j .

We also define $r_{u,v}^{min} = \min_{i \in G_u, j \in G_v} r_{i,j}$ to be the smallest distance between any two nodes of customer groups G_u and G_v .

For ease of notation, we define $d_{i,i} = r_{i,i} = 0$. We also make the following assumptions:

$$d_{i,j} \leq r_{i,j} \quad i \in X, j \in X, \quad (1)$$

$$d_{i,j} \leq d_{i,k} + d_{k,j} \quad i \in X, j \in X, k \in X, \quad (2)$$

$$r_{i,j} \leq r_{i,k} + d_{k,j} \quad i \in X, j \in X, k \in X. \quad (3)$$

Assumption (1) states that the direct cost is lower than or equal to the indirect cost, which is clearly the case in VRP settings. In the PCB setting, this is also the case as argued in [27]. Inequality (2) is the standard triangle inequality of the direct cost. Finally, (3) links direct and indirect costs. As such, it holds in VRP settings, and also in the PCB setting as shown in [27].

Note that (3) and (1) imply the triangle inequality of indirect cost. For future analysis, we rewrite (3) as

$$r_{i,j} \leq d_{i,i'} + r_{j,i'} \leq d_{i,i'} + r_{i',j'} + d_{j',j} \quad i \in X, i' \in X, j \in X, j' \in X. \quad (4)$$

2.1.2 Formal Definition of the Generalized and Expanded Multi-Cost Edge Network (*GEM-CEN*) Model

We also define a specially-formulated generalized and expanded multi-cost edge network formulation. This network is used in the cases when trucks are not required to be fully loaded. The *GEM-CEN* network is a complete undirected graph $\mathcal{G}(X, E)$ as before, where X is the set of nodes, and E is the set of edges. Let us also have G_1, G_2, \dots, G_K as a partition of X into K mutually exclusive and exhaustive groups as before. Likewise, every node i has a unique index $g(i) \in \{1, 2, \dots, K\}$ such that $i \in G_{g(i)}$. In this formulation $q + 1$

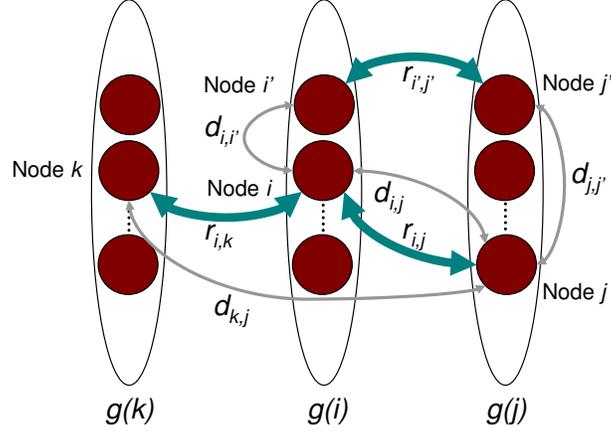


Figure 5: Generalized and expanded 2-cost edge network formulation

costs are defined for each edge, i.e., there is a direct cost and q indirect costs between any two nodes (see Figure 6). The indirect cost $r_{i,j}^h$ between node i and node j corresponds to the smallest indirect cost of going from node i to node j while visiting h depots en route, computed by solving a constrained shortest-path problem. While this might be computationally hard, in practice this limitation can be overcome by using abundant fast heuristics to solve such problems. We note that $r_{i,j}^1$ refers to the indirect cost between nodes i and j through a single depot. We also define $r_{u,v}^{h,min} = \min_{i \in G_u, j \in G_v} r_{i,j}^h$.

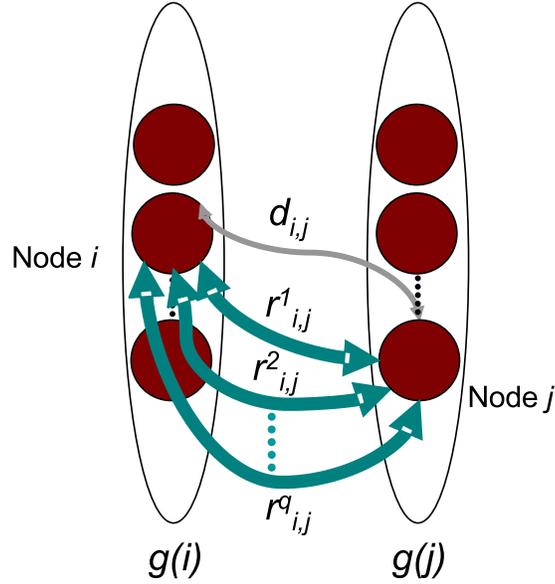


Figure 6: Generalized and expanded multi-cost edge network formulation

We make the following assumptions:

$$d_{i,j} \leq r_{i,j}^h \quad i \in X, j \in X, 1 \leq h \leq q, \quad (5)$$

$$d_{i,j} \leq d_{i,k} + d_{k,j} \quad i \in X, j \in X, k \in X, \quad (6)$$

$$r_{i,j}^h \leq r_{i,k}^h + d_{k,j} \quad i \in X, j \in X, k \in X, 1 \leq h \leq q, \quad (7)$$

$$r_{i,j}^{h_1} \leq r_{i,j}^{h_2} \quad i \in X, j \in X, 1 \leq h_1 \leq h_2 \leq q. \quad (8)$$

Assumptions (5)-(8) are similar to the already discussed assumptions (1)-(3). We derive

$$r_{i,j}^h \leq d_{i,i'} + r_{i',j'}^h + d_{j',j} \quad i \in X, i' \in X, j \in X, j' \in X \quad (9)$$

for every h . Condition (9) recognizes that the value of the shortest path through h depots is lower or equal to the value of the shortest path through $h + 1$ depots. In [27], we argue that this is the case for PCB manufacturing.

2.2 Variants of the Problem

We examine three specific variants of the problem. To the best of our knowledge, the GVRP, even by itself (without the generalizations motivated by the PCB setting), has not yet been studied from an analysis point of view.

The first variant assumes fully loaded trucks, the next allows partially loaded trucks, and the last variant takes the aspect of moving depots into account.

2.2.1 The Generalized Full-load Case

In this case, we assume that the vehicle always runs a full load on its way out from the depots. The problem is modeled based on *GE2 – CEN*. In other words, we assume that we fill the vehicle to capacity and the vehicle returns to a depot empty. We also assume that the depots are stationary. This is the most basic scenario that we analyze and it provides a building block for the remaining two scenarios.

Under the assumptions listed above, the problem being investigated here can be re-stated as that of finding a least cost GTSP-like (i.e., each node is visited at most once) sequence of edges in this specially formulated two-cost edge network. This problem differs from GTSP in that it uses a special cost function to compute the total cost of a GTSP-like ‘tour’. We use both the primary and secondary costs to compute the total cost, where the cost used on an edge depends upon the position of that edge in the tour. This special cost function is designed to reflect the actual vehicle routing type problem scenario.

Let us define a *segment* as a sequence of q consecutive edges on a GTSP-like tour. The cost of a segment is defined as the sum of the indirect cost $r_{i,j}$ of the first edge and the total direct costs $d_{i,j}$ of the last $q - 1$ consecutive edges of the segment (see Figure 7).

The GTSP-like tour we are interested in is a sequence of ζ consecutive segments of q edges each such that exactly one node in each G_1, \dots, G_K is visited.

The problem is to find the least cost GTSP-like tour $R = \{u_1, u_2, \dots, u_\zeta\}$ with segments u_1, \dots, u_ζ and the cost of the tour as determined by

$$C(R) = \sum_{s=1}^{\zeta} \left(r_{f(u_s)} + \sum_{e \in F(u_s)} d_e \right). \quad (10)$$

Given segment u , we denote by $f(u_s)$ the first edge in the segment, and by $F(u_s)$ the remaining edges in the segment. The orientation of each segment is defined with respect to the orientation of R .

2.2.2 The Generalized Partial-load Case

In the generalized partial-load case, we relax the assumption of the vehicle always carrying a full load from the depots.

The problem being investigated can thus be re-stated as that of finding the least cost GTSP-like sequence of edges in the *GEM – CEN*.

A *segment* is now defined to be a sequence of q or less consecutive edges on a GTSP-like tour.

The cost of a segment with h edges is now computed as the sum of $r_{i,j}^h$ of the first edge of the segment and the direct cost components $d_{i,j}$ of the last $h - 1$ consecutive edges of the segment.

Let us assume we have η segments in a GTSP-like tour with lengths h_1, \dots, h_η . The cost of the GTSP-like tour is now determined as

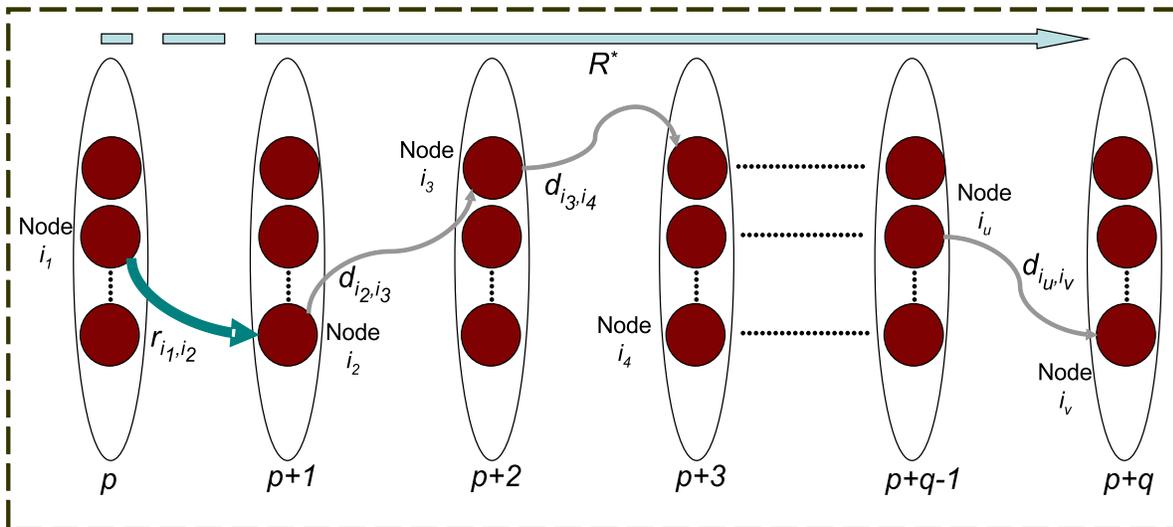


Figure 7: Cost of a segment for the generalized full-load case

$$C(R) = \sum_{s=1}^{\eta} \left(r_{f(u_s)}^{h_s} + \sum_{e \in F(u_s)} d_e \right).$$

2.2.3 The Generalized Partial-load Case with Waiting Cost

In this case, we relax the assumption of stationary depots and introduce the maximum component in the computation of the objective function. We assume that the depots are linked and move together. This is commonplace in most applications motivating such a setting, for example, robotics, automation, and railcars. An alternative way to state this assumption is to say that we are given a set of depots and their locations, and we have the flexibility of moving the entire ‘plane’ (all depots will move by the same amount). This captures the PCB setting since by moving the feeder all feeder slots move in sync by the same amount. Thus, there is a finite, discrete set of positions \mathcal{P} taken on by the depots in relation to the default “home” or base position.

We redefine the *GEM – CEN* to have nodes of the form (i, p) where i refers to a customer location and p refers to the position of the depots when the vehicle is at customer i . This makes a discrete approximation to the underlying continuous motion of the depots, but it is not anticipated to be a bad approximation in practical VRP settings. Note that the quality of the approximation can be controlled by expanding the set of all depot positions \mathcal{P} (clearly at the expense of a larger network). Then the problem being investigated can still be stated as that of finding the least cost GTSP-like sequence of edges in the *GEM – CEN*. However, the special cost function to compute the total cost of a GTSP-like tour is now modified to account for the moving depots. Note that the vehicle and the depots must synchronize at points of convergence, i.e., either the vehicle has to wait for the depot to move to a location, or the other way around. For the remainder of this paper, we continue to denote a customer node by a single index i assuming that the position information is implicitly stored in each node.

A *segment* is still defined to be a sequence of q or less consecutive edges on a GTSP-like tour, but the cost definition of a segment now includes a special cost between the two nodes corresponding to the start and end of a segment (see Figure 8). To this end, let $s_{i,j}$ be the special cost corresponding to the cost of moving the depots from positions defined by p_1 and p_2 corresponding to the nodes i and j , respectively. Values $s_{i,j}$ capture any waiting time penalties (in the PCB setting, the cost is actually measured in time units). The cost of a segment with h edges is now computed as the sum of $r_{i,k}^h$ of the first edge of the segment, and the maximum of the special cost $s_{i,j}$ between the start and end points of the segment and the direct cost

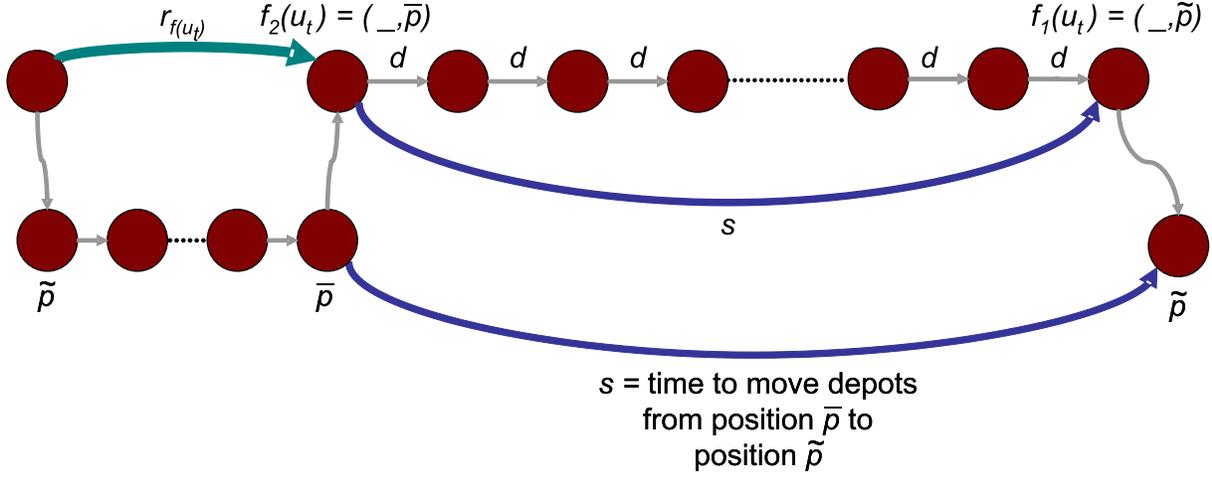


Figure 8: Cost of a segment for the generalized partial-load case with waiting cost

components $d_{k,m}$ of last $(h - 1)$ consecutive edges of the segment. The choice of the maximum operator captures the time/waiting interpretation of the cost. We note that again the segment is of length h rather than of length q .

For the partial load case with waiting cost, we need to impose the following additional assumption on the extent of the violation of the triangular inequality caused by the inclusion of the maximum component. We assume that

$$\frac{s_{i,j}}{\Delta} \leq d_{i,k} + d_{k,j} \quad i \in X, j \in X, k \in X \quad (11)$$

for a fixed finite value Δ . Requirement (11) states that s and d values might not satisfy the triangle inequality (if they do, there is no need for the maximum component), instead it can be violated by at most Δ .

Let us assume we have η segments $R = \{u_1, \dots, u_\eta\}$ in a GTSP-like tour with lengths h_1, \dots, h_η . The cost of the tour is now determined as

$$C(R) = \sum_{t=1}^{\eta} \left\{ r_{f(u_t)}^{h_t} + \max \left(\sum_{e \in F(u_t)} d_e, s_{f_2(u_t), f_1(u_t)} \right) \right\}$$

where we denote by $f_1(u_t)$ the head node of the last edge in the segment, and $f_2(u_t)$ the head node of the first edge in the segment.

3 Analysis of the Iterated Tour Partitioning Heuristic

Let us introduce the following notation.

$X = \{x_1, x_2, \dots, x_n\}$ is the set of nodes in the network

X_w = subset of nodes visited by the w^{th} segment of a GTSP-like tour

R^* = optimal GTSP-like tour over set X using $C(R^*)$ to determine its cost

G^* = optimal GTSP tour using direct costs on edges over set X with cost $\bar{C}(G^*) = \sum_{e \in G^*} d_e$

$\bar{r}_{G^*} = \sum_{a \in G^*} r_a / K$

$$\bar{r}_{G^*}^h = \sum_{a \in G^*} r_a^h / K$$

$$\bar{r}_{G^*}^{min} = \sum_{\{i,j\} \in G^*} r_{g(i),g(j)}^{min} / K$$

$$\bar{r}_{G^*}^{h,min} = \sum_{\{i,j\} \in G^*} r_{g(i),g(j)}^{h,min} / K$$

3.1 The Generalized Full-load Case

We start by formally describing the iterative tour partitioning heuristic. The heuristic first computes the optimal GTSP tour G^* by considering only direct cost d . Next a node and an orientation of G^* are fixed. We follow G^* based on the selected orientation by starting at the node and each $q - 1$ edges, we design the next edge to bear the indirect cost. After reaching back the original node we obtain a GTSP-like tour. The entire process is repeated for each possible starting node and the lowest cost solution among the q solutions is returned. (It actually suffices to only take q consecutive nodes as starting nodes.) The main result of this section is to establish a worst-case ratio for this heuristic.

Description of the Heuristic

1. Compute G^* .
2. Choose an arbitrary node p and an arbitrary orientation along G^* .
3. Partition G^* into ζ segments to get a solution starting from node p .
4. Record the underlying cost.
5. Repeat steps 2 - 4 by starting with customers $p + 1, p + 2, \dots, p + q - 1$ to obtain a solution with respect to each one of them.
6. Choose the best among the q solutions obtained and denote it by R^H .

3.1.1 Approximation Ratio

We start by establishing a lower bound on the optimal value.

Theorem A.1 We have

$$C(R^*) \geq \max \{ \bar{r}_{G^*}^{min}, \bar{C}(G^*) \}. \quad (12)$$

Proof: It is clear that we have $C(R^*) \geq \bar{C}(G^*)$ due to (1).

Let us first consider $\zeta > 1$. We also know that, for a given optimal GTSP tour G^* and given edge $e = \{k, l\} \in G^*$, both the tail customer group $G_{g(k)}$ and the head customer group $G_{g(l)}$ are visited in some order by the optimal solution R^* , albeit possibly through some different nodes, say, $i \in G_{g(k)}$ and $j \in G_{g(l)}$. We know that either one of the two possible paths in R^* between customer groups $G_{g(k)}$ and $G_{g(l)}$ must contain at least one edge that contributes the indirect cost since $\zeta \geq 2$. Let us renumber the nodes so that one path of the optimal solution between customer group $G_{g(k)}$ to $G_{g(l)}$ (along R^*) is denoted by $1, 2, \dots, \alpha, \beta, \dots, t$ with edge $\{\alpha, \beta\}$ contributing the indirect cost towards $C(R^*)$. Note that this might not be the only edge contributing the indirect cost. Similarly, let the nodes along the other path following R^* between customer group $G_{g(k)}$ and $G_{g(l)}$ be denoted by $1', 2', \dots, t'$. We have from our assumptions (see also Figure 9),

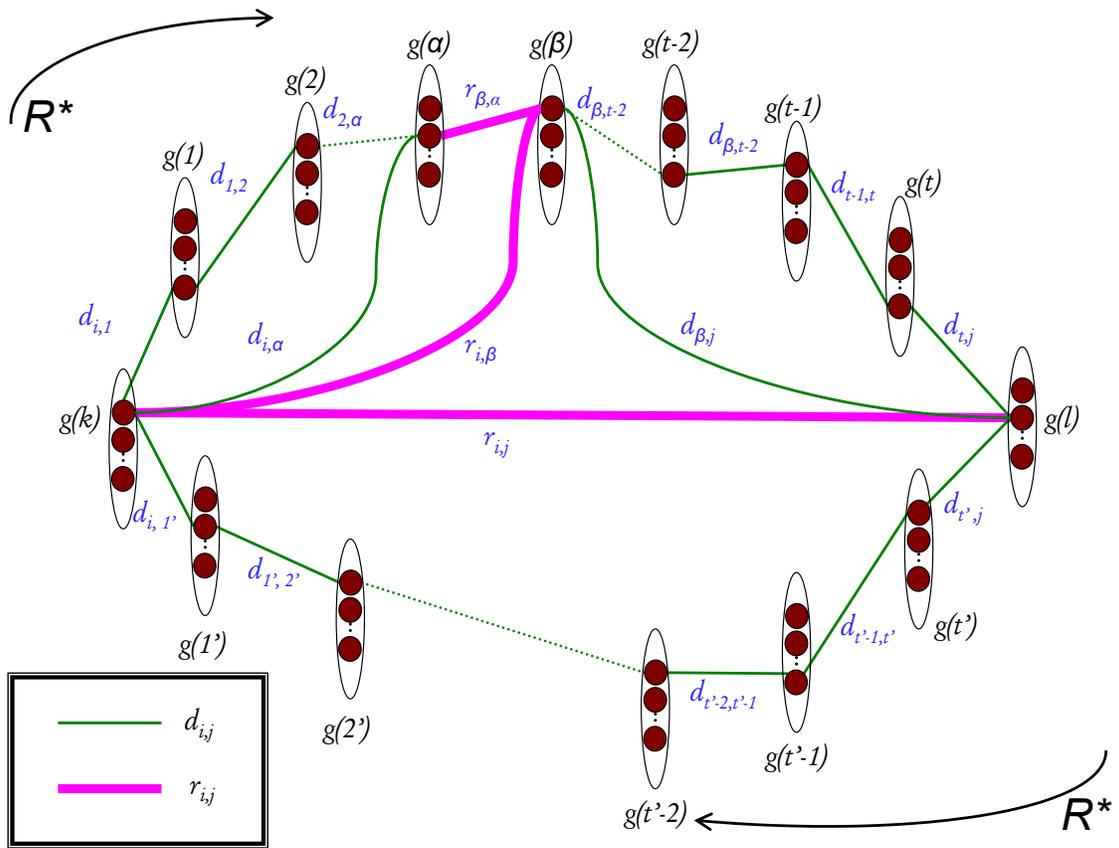


Figure 9: Derivation of lower bound

$$r_e^{min} = r_{g(k),g(l)}^{min} \leq r_{i,j} \tag{13}$$

$$\leq r_{i,\beta} + d_{\beta,j} \tag{14}$$

$$\leq r_{i,\beta} + (d_{t,j} + d_{t-1,t} + d_{t-2,t-1} + d_{\beta,t-2}) \tag{15}$$

$$\leq d_{i,\alpha} + r_{\alpha,\beta} + (d_{t,j} + d_{t-1,t} + d_{t-2,t-1} + d_{\beta,t-2}) \tag{16}$$

$$\leq (d_{i,1} + d_{1,2} + d_{\alpha-1,\alpha}) + r_{\alpha,\beta} + (d_{t,j} + d_{t-1,t} + d_{t-2,t-1} + d_{\beta,t-2}) \tag{17}$$

$$\leq C(R^*),$$

where (14) and (16) follow from (2), (13) and (15) follow from (3), and (17) follows from (1) and the nonnegativity of costs.

If $\zeta = 1$ and $r_{i,j}$ is the only indirect cost edge in R^* , we clearly have $r_{i,j} \leq C(R^*)$. If $\zeta = 1$ and $r_{i,j}$ is not the only indirect cost edge, it implies that there is a different edge on R^* contributing its indirect cost and thus we proceed as in the case $\zeta \geq 2$.

Since $r_e \leq C(R^*)$ for every $e \in G^*$, it follows that $\bar{r}_{G^*}^{min} \leq C(R^*)$. □

Next we obtain an upper bound on $C(R^*)$. Let R^H be the GTSP-like tour solution produced by the heuristic with $C(R^H)$ as the underlying cost.

Theorem B.1 We have

$$C(R^*) \leq C(R^H) \leq \frac{K}{q} \bar{r}_{G^*} + \left(1 - \frac{1}{q}\right) \bar{C}(G^*). \tag{18}$$

Proof: We observe that the cumulative length of the q solutions encountered by the heuristic is equal to

$$\sum_{e \in G^*} r_e + (q - 1) \cdot \bar{C}(G^*).$$

The value $C(R^H)$ equals the best solution value among the q solutions, and thus must be less than or equal to the average value of all encountered solutions. Therefore, we conclude

$$C(R^H) \leq \frac{\sum_{e \in G^*} r_e}{q} + \left(1 - \frac{1}{q}\right) \bar{C}(G^*),$$

or equivalently

$$C(R^H) \leq \frac{K}{q} \bar{r}_{G^*} + \left(1 - \frac{1}{q}\right) \bar{C}(G^*).$$

The statement $C(R^*) \leq C(R^H)$ clearly holds. □

From Theorem A.1 and Theorem B.1, we obtain the following main result.

Theorem C.1 We have

$$\frac{C(R^H)}{C(R^*)} \leq \frac{\bar{r}_{G^*}}{\max\{\bar{r}_{G^*}^{min}, \bar{C}(G^*)\}} \frac{K}{q} + \frac{\bar{C}(G^*)}{\max\{\bar{r}_{G^*}^{min}, \bar{C}(G^*)\}} \left(1 - \frac{1}{q}\right). \tag{19}$$

If we reduce the problem to a standard VRP scenario where there are no clusters and there is no need for the truck to visit several depots between two routes, we observe that the above derived bound reduces to the bounds in [15] and [20].

3.1.2 Tightness of the Bound

In this section, we prove that the derived bound (19) is tight by finding a family of instances with K clusters for which (19) is at equality.

Consider network $\mathcal{G}(X, E)$ with K customer groups, m nodes each. Consider one node, called the ‘leader’ node from each of the K groups being placed at equal intervals along the perimeter of a circle of diameter ϕ and consecutive nodes being placed at angular intervals of $\epsilon = \pi/n$ incrementally from the leader node of the underlying group (see Figure 10). Thus, we have a total of $n = K \cdot m$ nodes and we assume that q divides n . Note that by definition, we have $\epsilon \leq 2\pi/n$ and thus the groups do not overlap. We fix a given orientation of the circle. Let $d_{i,j}$ be the distance along the circle from customer i to customer j and $r_{i,j} = d_{i,j} + \delta$ where $\delta = q/K^2$. Then the optimal GTSP tour follows the circle itself using, say, the first node or leader from each customer group. We have $\bar{C}(G^*) = \pi\phi$.

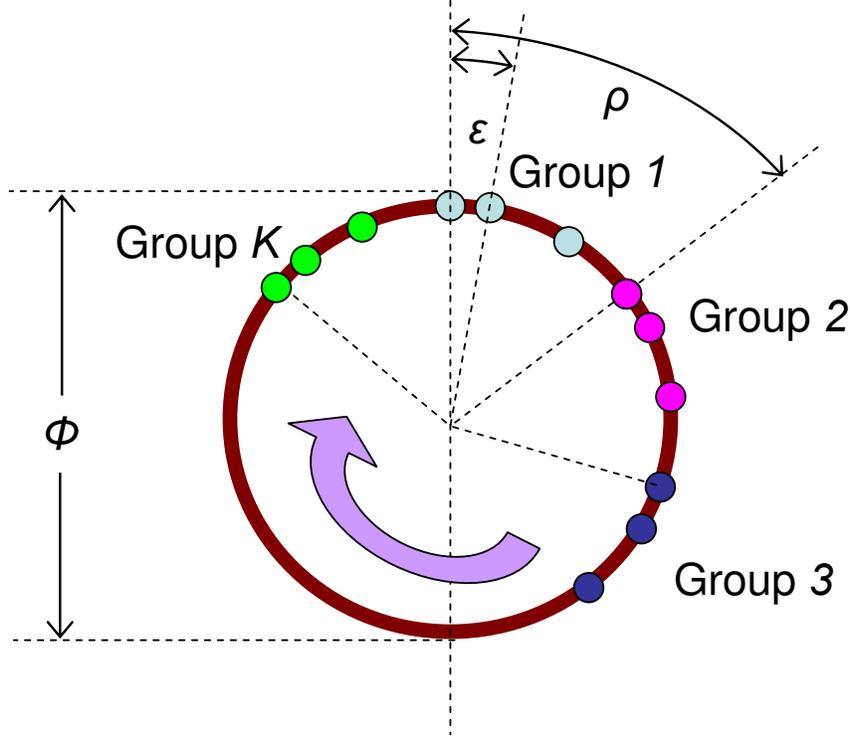


Figure 10: Tightness proof instance

In our example, we clearly satisfy the required assumption $r_{i,j} \geq d_{i,j}$. Furthermore, we also satisfy the other two assumptions $d_{i,j} \leq d_{i,k} + d_{k,j}$ and $r_{i,j} \leq r_{i,k} + d_{k,j}$ (because $\delta + d_{i,j} = r_{i,j}$). We know that

$$d_{i,j} = \frac{\phi}{2} (\rho + (b - a)\epsilon)$$

for all i, j belonging to consecutive groups on the circle where $\rho = 2\pi/K$ and a, b are the sequence numbers of nodes i, j in their respective groups. We note that $r_{u,v}^{min} = (\rho - m\epsilon)\phi/2 + \delta$ for two consecutive clusters u and v . Therefore, we have

$$\bar{r}_{G^*}^{min} = \frac{1}{K} \left\{ K \frac{\rho\phi}{2} - Km \frac{\epsilon\phi}{2} + K\delta \right\} = \frac{\rho\phi}{2} - \frac{m\epsilon\phi}{2} + \delta.$$

On the other hand, $\bar{r}_{G^*} = (\pi\phi + K\delta)/K = \rho\phi/2 + \delta$.

Solution R^* visits the nodes in the same sequence along the perimeter of the circle, and its value is clearly $\frac{K\rho\phi}{2} + \frac{n}{q} \cdot \delta$ since r values are selected n/q times.

The heuristic begins with the GTSP tour and chooses, regardless of the starting point due to symmetry, a solution of equal cost, i.e., $C(R^H) = \frac{K\rho\phi}{2} + \frac{n}{q} \cdot \delta$. We conclude $C(R^H)/C(R^*) = 1$. We also obtain, by

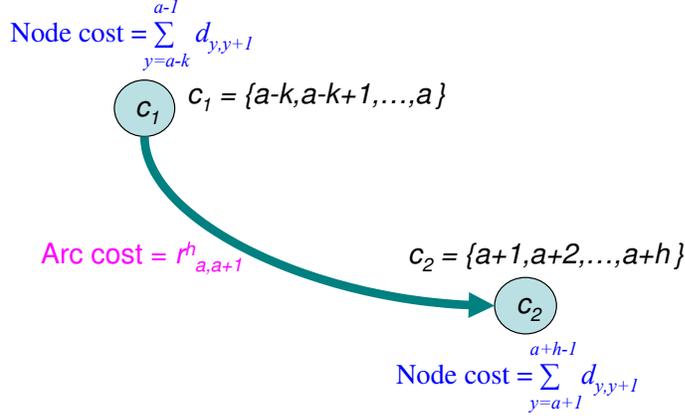


Figure 11: Costs related to the auxiliary network

substituting values derived above

$$\frac{\bar{r}_{G^*}}{\max\{\bar{r}_{G^*}^{\min}, \bar{C}(G^*)\}} \frac{K}{q} + \frac{\bar{C}(G^*)}{\max\{\bar{r}_{G^*}^{\min}, \bar{C}(G^*)\}} \left(1 - \frac{1}{q}\right) = 1 + \frac{1}{K\pi\phi}.$$

Here we use the definition of ϵ and δ , which imply that $\max\{\bar{r}_{G^*}^{\min}, \bar{C}(G^*)\} = \bar{C}(G^*)$. This can be seen by a straightforward calculation. Hence, as $K \rightarrow \infty$, this shows that the bound is tight. \square

3.2 The Generalized Partial-load Case

In this section, we perform the analysis for the case when the vehicle can carry less than q units. The major challenge is to adapt the iterated tour partitioning heuristic. While the heuristic still finds a feasible solution, the resulting encountered set of feasible solutions is too stringent (only full truck loads are considered). For this reason we need to generalize it. The act of finding G^* remains the same.

As before, we fix the orientation of G^* . For simplicity, we label the nodes $1, \dots, K$ along G^* . Let us create a new auxiliary network where there is a node for every set of q or less consecutive edges on G^* . The number of nodes is $\binom{K}{q}$.

Two nodes c_1 and c_2 in this new auxiliary network are connected by an arc with cost $r^h_{i,j}$ if the tail node c_1 represents a sequence of edges (in the original network) ending with node i and the head node c_2 represents a sequence of h edges (in the original network) starting with node j (see Figure 11).

A node itself carries an additional cost equivalent to the cumulative cost of traversing the sequence of arcs that it represents in the original network using the direct costs. Thus, if node c_1 in the auxiliary network represents the sequence of nodes $a-k, a-k+1, \dots, a$, the cost of the node in the auxiliary network is defined by $\sum_{y=a-k}^{a-1} d_{y,y+1}$.

Let us now fix a customer group p . To this auxiliary network, we add a source node s and sink node t to facilitate the computation of the shortest path through the network. Source s is connected to each node corresponding to a sequence originating at customer group p . The cost of an (s, c_1) arc equals to the cost of picking up as many items as nodes in the original network represented by c_1 , and then moving to node p . Similarly, each node corresponding to a sequence ending with p is connected to t with a cost of 0.

We repeat the construction of the network starting with customer groups $p+1, p+2, \dots, p+q-1$ to obtain q such networks. It is easy to see that each $s-t$ path in these networks yields a feasible solution. In addition, each solution following G^* corresponds to an $s-t$ path in one of the networks. By the definition of costs, the cost of a path in any one of these auxiliary networks equals to the cost of the corresponding GTSP-like solution. We note that the presented methodology of finding the shortest $s-t$ path on an auxiliary network is akin to the methodology used by Beasley [3], however, we extend it by 1) using q such networks, and 2) modeling both arc and node costs, due to the more complicated nature of the problem at hand.

The formal description of the heuristic follows.

Description of the Heuristic

1. Compute G^* .
2. Choose an arbitrary customer group p and an arbitrary orientation along G^* .
3. Construct the q auxiliary networks.
4. Compute the shortest $s - t$ path in each one of the created auxiliary networks.
5. Choose the best among the q solutions so obtained and denote it by R^H .

3.2.1 Approximation Ratio

We start by generalizing Theorem A.1

Theorem A.2 We have

$$C(R^*) \geq \max \left\{ r_{G^*}^{1, \min}, \bar{C}(G^*) \right\}. \quad (20)$$

Proof: It is clear that we have $C(R^*) \geq \bar{C}(G^*)$ due to (5).

Let first $\zeta > 1$. We know that, for a given optimal GTSP tour G^* and given edge $e = \{k, l\} \in G^*$, both the tail customer group $G_{g(k)}$ and the head customer group $G_{g(l)}$ are visited in some sequence on the optimal solution R^* , albeit possibly through some different nodes, say, i and j . We know that either one of the two possible paths in R^* between customer groups $G_{g(k)}$ and $G_{g(l)}$ must contain at least one edge which contributes the indirect cost since $\zeta > 1$.

Let us renumber the nodes so that nodes $1, 2, \dots, \alpha_1, \beta_1, \dots, \alpha_\tau, \beta_\tau, \dots, t$ form one path from node i to node j along R^* with τ edges $\{\alpha_1, \beta_1\}, \dots, \{\alpha_\tau, \beta_\tau\}$ contributing indirect costs $r_{\alpha_1, \beta_1}^{p_1}, \dots, r_{\alpha_\tau, \beta_\tau}^{p_\tau}$, and nodes $1', 2', \dots, \alpha_{1'}, \beta_{1'}, \dots, \alpha_{\tau'}, \beta_{\tau'}, \dots, t'$ form the other path with τ' edges $\{\alpha_{1'}, \beta_{1'}\}, \dots, \{\alpha_{\tau'}, \beta_{\tau'}\}$ contributing indirect costs $r_{\alpha_{1'}, \beta_{1'}}^{p_{1'}}, \dots, r_{\alpha_{\tau'}, \beta_{\tau'}}^{p_{\tau'}}$. We have (see Figure 12)

$$\begin{aligned} C(R^*) = & \left\{ d_{i,1} + d_{1,2} + \dots + r_{\alpha_1, \beta_1}^{p_1} + \dots \right. \\ & \left. + r_{\alpha_2, \beta_2}^{p_2} + \dots + r_{\alpha_\tau, \beta_\tau}^{p_\tau} + \dots + d_{t,j} \right\} \\ & + \left\{ d_{j,t'} + \dots + r_{\alpha_{\tau'}, \beta_{\tau'}}^{p_{\tau'}} + \dots + r_{\beta_{2'}, \alpha_{2'}}^{p_{2'}} + \dots \right. \\ & \left. + r_{\beta_{1'}, \alpha_{1'}}^{p_{1'}} + \dots + d_{1',i} \right\} \end{aligned} \quad (21)$$

$$\begin{aligned} \geq & \left\{ d_{i, \alpha_1} + r_{\alpha_1, \beta_1}^{p_1} + d_{\beta_1, \alpha_2} + r_{\alpha_2, \beta_2}^{p_2} + d_{\beta_2, \alpha_3} + \dots \right. \\ & \left. + r_{\alpha_\tau, \beta_\tau}^{p_\tau} + d_{\beta_\tau, t} + d_{t,j} \right\} \\ & + \left\{ d_{j,t'} + d_{t', \beta_{\tau'}} + r_{\beta_{\tau'}, \alpha_{\tau'}}^{p_{\tau'}} + \dots \right. \\ & \left. + r_{\beta_{2'}, \alpha_{2'}}^{p_{2'}} + d_{\alpha_{2'}, \beta_{1'}} + r_{\beta_{1'}, \alpha_{1'}}^{p_{1'}} + d_{\alpha_{1'}, 1'} + d_{1',i} \right\} \end{aligned} \quad (22)$$

$$\begin{aligned} \geq & \left\{ d_{i, \alpha_1} + r_{\alpha_1, \beta_1}^1 + d_{\beta_1, \alpha_2} + r_{\alpha_2, \beta_2}^1 \right. \\ & \left. + d_{\beta_2, \alpha_3} + \dots + r_{\alpha_\tau, \beta_\tau}^1 + d_{\beta_\tau, t} + d_{t,j} \right\} \\ & + \left\{ d_{j,t'} + d_{t', \beta_{\tau'}} + r_{\alpha_{\tau'}, \beta_{\tau'}}^1 + \dots + r_{\beta_{2'}, \alpha_{2'}}^1 \right. \\ & \left. + d_{\alpha_{2'}, \beta_{1'}} + r_{\beta_{1'}, \alpha_{1'}}^1 + d_{\alpha_{1'}, 1'} + d_{1',i} \right\} \end{aligned} \quad (23)$$

$$\geq r_{i,j}^1 \quad (24)$$

$$\geq r_{g(k), g(l)}^{1, \min}$$

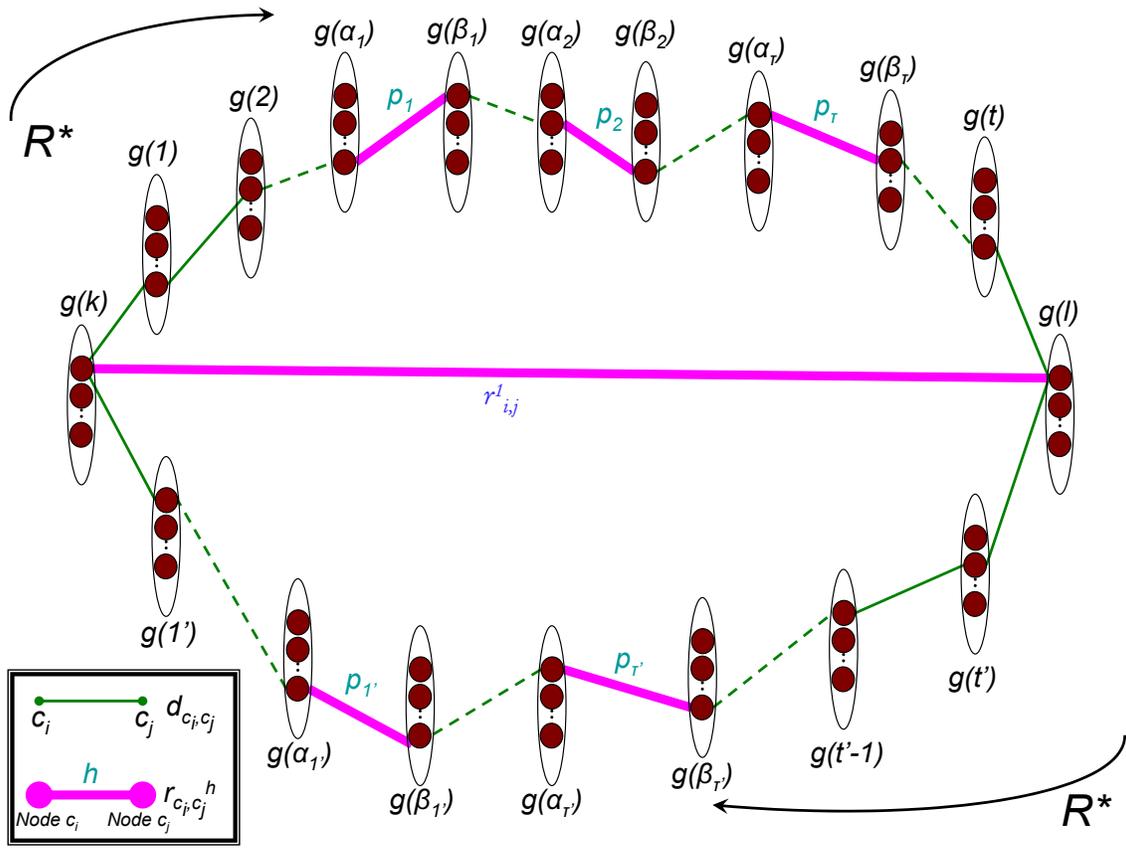


Figure 12: Derivation of modified lower bound

where (21) follows from the definition of $C(R^*)$, (22) follows from (6), (23) follows from (8), and (24) follows from (6) and (7).

If $\zeta = 1$ and $r_{i,j}^h$ is the only indirect cost edge in R^* , we clearly have $r_{i,j}^1 \leq C(R^*)$. If $\zeta = 1$ and $r_{i,j}^h$ is not the only indirect cost edge, it implies that there is a different edge on R^* contributing its indirect cost and thus we proceed as in the case $\zeta \geq 2$.

Since $r_{g(a),g(b)}^{1,min} \leq C(R^*)$ for every $\{a, b\} \in G^*$, it follows that $\bar{r}_{G^*}^{1,min} \leq C(R^*)$. \square

Next, we establish an upper bound on $C(R^H)$.

Theorem B.2 For each $h = 1, 2, \dots, q$ and $K \bmod h = 0$, we have

$$C(R^*) \leq C(R^H) \leq \frac{K}{h} \bar{r}_{G^*}^h + \left(1 - \frac{1}{h}\right) \bar{C}(G^*). \quad (25)$$

For each $h = 1, 2, \dots, q$ and $K \bmod h > 0$, we have

$$C(R^*) \leq C(R^H) \leq \frac{2K}{h} \bar{r}_{G^*}^h + \left(1 - \frac{2}{h}\right) \bar{C}(G^*). \quad (26)$$

Proof: We observe that each of the q solutions is a shortest $s - t$ path in the respective network. Thus, it is shorter than any other path through the underlying network.

Suppose we only consider paths such that each node on the path represents a sequence of h customer groups in the original network except for the very last node, which could correspond to a sequence of length less than h . We derive that there are only h such unique paths among the q networks denoted by $\mathcal{P}_1, \dots, \mathcal{P}_h$, and with lengths $L(\mathcal{P}_1), \dots, L(\mathcal{P}_h)$, respectively.

Let $\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_q$ be the shortest paths through the q networks and $L(\mathcal{S}_1), L(\mathcal{S}_2), \dots, L(\mathcal{S}_q)$ be their lengths.

Imagine that we have $q \cdot h$ networks where each of the q networks is repeated h times. We thus have q repetitions of each of the paths $\mathcal{P}_1, \dots, \mathcal{P}_h$ and h repetitions of paths corresponding to $\mathcal{S}_1, \dots, \mathcal{S}_q$ among these networks. Since $L(\mathcal{S}_i)$ is a shortest path value, it follows that $L(\mathcal{S}_i) \leq L(\mathcal{P}_k)$ for each matching k . We thus have

$$h \cdot \sum_{i=1}^q L(\mathcal{S}_i) \leq q \cdot \sum_{j=1}^h L(\mathcal{P}_j).$$

Therefore, we conclude that

$$\frac{\sum_{i=1}^q L(\mathcal{S}_i)}{q} \leq \frac{\sum_{j=1}^h L(\mathcal{P}_j)}{h}.$$

Clearly, the value $C(R^H)$, which is equal to the best solution value among the q solutions, must be less than or equal to the average value of these q solutions. Therefore, we have

$$C(R^H) \leq \frac{\sum_{i=1}^q L(\mathcal{S}_i)}{q} \leq \frac{\sum_{j=1}^h L(\mathcal{P}_j)}{h}.$$

Let first $K \bmod h = 0$. We know that

$$\sum_{j=1}^h L(\mathcal{P}_j) = \sum_{e \in G^*} r_e^h + (h-1) \bar{C}(G^*)$$

because of the structure of the chosen paths. Since $K \bmod h = 0$, each vehicle dispatched in \mathcal{P}_j carries exactly K/h units. Therefore, we get

$$C(R^H) \leq \frac{\sum_{e \in G^*} r_e^h}{h} + \left(1 - \frac{1}{h}\right) \bar{C}(G^*)$$

or

$$C(R^H) \leq \frac{K}{h} \bar{r}_{G^*}^h + \left(1 - \frac{1}{h}\right) \bar{C}(G^*).$$

Let now $K \bmod h > 0$. In this case, the vehicles do not dispatch the same number of units. It is not difficult to see that

$$\sum_{j=1}^h L(\mathcal{P}_j) = \sum_{e \in E_1} (r_e^h + (h-1) \cdot d_e) + \sum_{e \in E_2} (r_e^h + r_e^{\bar{h}} + (h-2) \cdot d_e)$$

for $\bar{h} = K \bmod h - 1 < h$. Here $G^* = E_1 \cup E_2, E_1 \cap E_2 = \emptyset$. Since $r_e^{\bar{h}} \leq r_e^h$ and $d_e \leq r_e^h$ for every e , we obtain

$$\begin{aligned} \sum_{j=1}^h L(\mathcal{P}_j) &\leq \sum_{e \in E_1} (2r_e^h + (h-2) \cdot d_e) + \sum_{e \in E_2} (2r_e^h + (h-2) \cdot d_e) \\ &= \sum_{e \in G^*} (2r_e^h + (h-2) \cdot d_e) = 2 \sum_{e \in G^*} r_e^h + (h-2) \cdot \bar{C}(G^*). \end{aligned}$$

The rest of the proof follows the steps from the case $K \bmod h = 0$. Inequality $C(R^*) \leq C(R^H)$ is trivially true. \square

Since $q \leq K$, we note that if $1 \leq h \leq q-1$, then $K \bmod h > 0$, but if $h = q$, then $K \bmod h$ can either be 0 or positive. From Theorem A.2 and Theorem B.2, and defining

$$\mathcal{H}(\alpha) = \begin{cases} 2 \frac{\bar{r}_{G^*}^\alpha}{\max\{\bar{r}_{G^*}^{1,min}, \bar{C}(G^*)\}} \frac{K}{\alpha} + \frac{\bar{C}(G^*)}{\max\{\bar{r}_{G^*}^{1,min}, \bar{C}(G^*)\}} \left(1 - \frac{2}{\alpha}\right) & K \bmod \alpha > 0 \\ \frac{\bar{r}_{G^*}^\alpha}{\max\{\bar{r}_{G^*}^{1,min}, \bar{C}(G^*)\}} \frac{K}{\alpha} + \frac{\bar{C}(G^*)}{\max\{\bar{r}_{G^*}^{1,min}, \bar{C}(G^*)\}} \left(1 - \frac{1}{\alpha}\right) & K \bmod \alpha = 0, \end{cases}$$

we obtain the following main result.

Theorem C.2 We have

$$\frac{C(R^H)}{C(R^*)} \leq \min_{\alpha \in \{1, 2, \dots, q\}} \mathcal{H}(\alpha). \quad (27)$$

Since both [15] and [20] did not consider the partial load case, we cannot compare bound (27) to their results.

3.2.2 Tightness of the Bound

In this section, we prove that the modified bound (27) is tight by finding a family of instances with K customer groups, each with m nodes, for which (27) is at equality.

We consider the same example as in Section 3.1.2, but with q indirect-cost edges between every pair of nodes. We let $r_{a,b}^h = d_{a,b} + \delta$ be the distance along the circle from customer a to customer b for $1 \leq h \leq q$ with an added term δ . All assumptions (5)-(8) are clearly satisfied. Then the optimal GTSP tour follows the circle itself and $\bar{C}(G^*) = \pi\phi$.

As in Section 3.1.2, we have

$$\bar{r}_{G^*}^{h,min} = \frac{\rho\phi}{2} - \frac{m\epsilon\phi}{2} + \delta.$$

We obtain a series of optimal solutions of equal objective value by visiting the nodes in the same sequence along the perimeter of the circle. An optimal solution R^* also is using the indirect cost $r_{i,j}^1$ on each edge $\{i, j\}$ and is following the optimal GTSP tour. Thus, the objective value of the solution is clearly $C(R^*) = K\rho\phi/2 + n/q \cdot \delta$ regardless of whether $K \bmod h = 0$ or $K \bmod h > 0$.

The heuristic begins with the TSP tour and chooses, regardless of the starting point, a solution of equal cost, i.e., $C(R^H) = \frac{K\rho\phi}{2} + \frac{n}{q} \cdot \delta$ again regardless of whether $K \bmod h = 0$ or $K \bmod h > 0$. We conclude $C(R^H)/C(R^*) = 1$ in either scenario.

If $K \bmod h = 0$, we obtain, by substituting values obtained above

$$\mathcal{H}(h) = \frac{1}{K} \cdot \frac{K}{h} + \left(1 - \frac{1}{h}\right) + \frac{1}{K\pi\phi} = 1 + \frac{1}{K\pi\phi}$$

and when $K \bmod h > 0$, likewise we obtain

$$\mathcal{H}(h) = 2 \cdot \frac{1}{K} \cdot \frac{K}{h} + \left(1 - \frac{2}{h}\right) + \frac{1}{K\pi\phi} = 1 + \frac{1}{K\pi\phi}$$

Hence, as $K \rightarrow \infty$, the bound can be seen to be tight. \square

3.3 The Generalized Partial-load Case with Waiting Cost

In this section, we perform the analysis for the case when the depots can move in the plane, and thereby a waiting cost is introduced into the cost computation function for the routes made by a vehicle. We create a new auxiliary network where there is a node for every set of q or less consecutive edges on G^* as before. However, in this network, we modify the node costs to be the maximum of the special cost and the cumulative cost of traversing the sequence of edges that the head node represents in the original network using the direct costs (see Figure 13).

We again fix a customer group p and add a source node s and sink node t to facilitate the computation of the shortest path through the network as before. Again, as before, source s is connected to each node corresponding to a sequence originating at group p . Similarly, each node corresponding to a sequence ending with p is connected to t .

We repeat the construction of the network starting with nodes $p+1, p+2, \dots, p+q-1$ to obtain q such networks. It is easy to see that each $s-t$ path in these networks yields a feasible solution. In addition, each solution following T^* corresponds to an $s-t$ path in one of the networks.

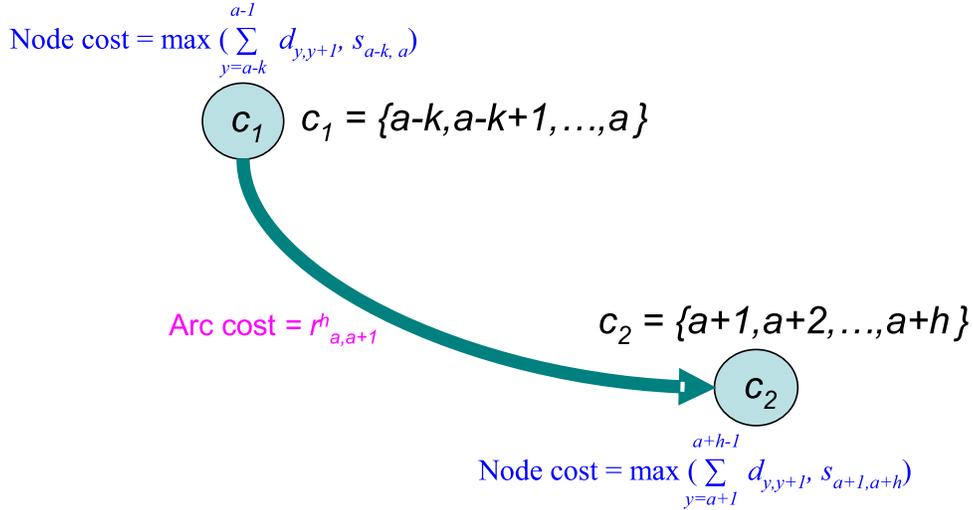


Figure 13: Costs related to the auxiliary network

The heuristic is exactly the same as before except using these networks with new definitions of costs. Also in this case we are able to analyze the heuristic.

Theorem A.3 We have, as before

$$C(R^*) \geq \max \left\{ r_{G^*}^{1, \min}, \bar{C}(G^*) \right\}. \quad (28)$$

Proof: It is clear from (12) that $C(R^*)$ in this case is only going to take on a higher value than the $C(R^*)$ computed in Theorem A.2. Since the conditions and assumptions of Theorem A.2 continue to hold here as well, this completes the proof. \square

To upper bound the cost of the heuristic, we obtain the following result.

Theorem B.3 For each $h = 1, 2, \dots, q$ and $K \bmod h = 0$, we have

$$C(R^*) \leq C(R^H) \leq \frac{K}{h} \bar{r}_{G^*}^h + \Delta \left(1 - \frac{1}{h}\right) \bar{C}(G^*). \quad (29)$$

For each $h = 1, 2, \dots, q$ and $K \bmod h > 0$, we have

$$C(R^*) \leq C(R^H) \leq \frac{2K}{h} \bar{r}_{G^*}^h + \Delta \left(1 - \frac{2}{h}\right) \bar{C}(G^*). \quad (30)$$

Proof: For any two nodes i, j and path \mathcal{P} connecting these two nodes, from (11) and (6) we obtain $s_{i,j} \leq \Delta \sum_{e \in \mathcal{P}} d_e$. In turn it follows that $\max\{s_{i,j}, \sum_{e \in \mathcal{P}} d_e\} \leq \Delta \sum_{e \in \mathcal{P}} d_e$.

We observe as before that each of the q solutions is a shortest $s-t$ path in the respective network. Thus, it is shorter than any other path through the underlying network.

Suppose we again only consider paths such that each node on the path represents a sequence of h groups in the original network except for the very last node, which could correspond to a sequence of length less than h . We derive as before that there are only h such unique paths among the q networks denoted by $\mathcal{P}_1, \dots, \mathcal{P}_h$ with lengths denoted by $L(\mathcal{P}_1), \dots, L(\mathcal{P}_h)$. We know from the proof of Theorem B.2 that

$$\frac{\sum_{i=1}^q L(\mathcal{S}_i)}{q} \leq \frac{\sum_{j=1}^h L(\mathcal{P}_j)}{h}$$

and therefore, we have

$$C(R^H) \leq \frac{\sum_{i=1}^q L(\mathcal{S}_i)}{q} \leq \frac{\sum_{j=1}^h L(\mathcal{P}_j)}{h}.$$

Let first $K \bmod h = 0$. We know that

$$\sum_{j=1}^h L(\mathcal{P}_j) \leq \sum_{e \in G^*} r_e^h + \Delta \cdot (h-1) \bar{C}(G^*)$$

because of our assumption (11) and the structure of the chosen paths.

Let now $K \bmod h > 0$. In this case, as in the proof of Theorem B.2 the vehicles do not dispatch the same number of units and we obtain

$$\sum_{j=1}^h L(\mathcal{P}_j) \leq \sum_{e \in E_1} (r_e^h + \Delta \cdot (h-1) \cdot d_e) + \sum_{e \in E_2} (r_e^h + r_e^{\bar{h}} + \Delta \cdot (h-2) \cdot d_e).$$

The rest of the proof is identical to the proof of Theorem B.2. \square

From Theorem A.3 and Theorem B.3, and defining

$$\mathcal{H}(\alpha) = \begin{cases} 2 \frac{\bar{r}_{G^*}^\alpha}{\max\{\bar{r}_{G^*}^{1, \min}, \bar{C}(G^*)\}} \frac{K}{\alpha} + \frac{\bar{C}(G^*)}{\max\{\bar{r}_{G^*}^{1, \min}, \bar{C}(G^*)\}} \Delta \cdot \left(1 - \frac{2}{\alpha}\right) & K \bmod \alpha > 0 \\ \frac{\bar{r}_{G^*}^\alpha}{\max\{\bar{r}_{G^*}^{1, \min}, \bar{C}(G^*)\}} \frac{K}{\alpha} + \frac{\bar{C}(G^*)}{\max\{\bar{r}_{G^*}^{1, \min}, \bar{C}(G^*)\}} \Delta \cdot \left(1 - \frac{1}{\alpha}\right) & K \bmod \alpha = 0, \end{cases}$$

we obtain the following main result.

Theorem C.3 We have

$$\frac{C(R^H)}{C(R^*)} \leq \min_{\alpha \in \{1, 2, \dots, q\}} \mathcal{H}'(\alpha). \quad (31)$$

3.4 Results for Standard VRP

In this section, we summarize the results when the problem is reduced to the standard multi-depot VRP case. Recall that cost $r_{i,j}$ is defined as the cost of the truck finishing the last delivery at node i , next visiting q depots (not modeled explicitly), and then moving to node j with q items to start the next delivery, in the most cost effective way. They are obtained from the generalized cases by setting m , the number of nodes in each group to be equal to 1.

In the full-load case, [20], we obtain the following known results. Since this refers to the standard VRP case, we have no partitions G_i , and thus, by shrinking each partition to contain only a single node, we can replace G^* with its equivalent of T^* defined as

$$T^* = \text{optimal TSP tour using direct costs on edges over set } X \text{ with cost } \bar{C}(T^*) = \sum_{e \in T^*} d_e.$$

Theorem A.4 We have $m = 1, G^* = T^*$, and $K = n$. This gives us the following results:

$$\begin{aligned} C(R^*) &\geq \max\{\bar{r}_{T^*}, \bar{C}(T^*)\}, \\ C(R^*) &\leq C(R^H) \leq \frac{\sum_{e \in T^*} r_e}{q} + \left(1 - \frac{1}{q}\right) \bar{C}(T^*), \\ \frac{C(R^H)}{C(R^*)} &\leq \frac{\bar{r}_{T^*}}{\max\{\bar{r}_{T^*}, \bar{C}(T^*)\}} \frac{n}{q} + \frac{\bar{C}(T^*)}{\max\{\bar{r}_{T^*}, \bar{C}(T^*)\}} \left(1 - \frac{1}{q}\right). \end{aligned} \quad (32)$$

A bound independent of T^* reads

$$\frac{C(R^H)}{C(R^*)} \leq \frac{n-1}{q} + 1. \quad (33)$$

Inequality (33) follows from (32) due to

$$\frac{\bar{r}_{T^*}}{\max\{\bar{r}_{T^*}, \bar{C}(T^*)\}} \leq 1, \quad \frac{\bar{C}(T^*)}{\max\{\bar{r}_{T^*}, \bar{C}(T^*)\}} \leq 1.$$

The partial-load case has not been studied previously and thus the results below are new. Here $r_{i,j}^h$ is interpreted as the indirect cost of going from node i to node j while visiting h depots en route in the most cost-efficient manner. Again, we can substitute G^* with T^* due to the underlying absence of partitions as a result of setting $m = 1$.

Theorem A.5 We have

$$C(R^*) \geq \max\{\bar{r}_{T^*}^1, \bar{C}(T^*)\}.$$

For each $h = 1, 2, \dots, q$, if $n \bmod h = 0$, then

$$C(R^*) \leq C(R^H) \leq \frac{n}{h} \bar{r}_{T^*}^h + \left(1 - \frac{1}{h}\right) \bar{C}(T^*),$$

and if $n \bmod h > 0$,

$$C(R^*) \leq C(R^H) \leq \frac{2n}{h} \bar{r}_{T^*}^h + \left(1 - \frac{2}{h}\right) \bar{C}(T^*).$$

We also have

$$\frac{C(R^H)}{C(R^*)} \leq \min_{\alpha \in \{1, 2, \dots, q\}} \mathcal{H}(\alpha).$$

A bound independent of T^* reads

$$\frac{C(R^H)}{C(R^*)} \leq \frac{n-1}{q} + 1$$

if $K \bmod q = 0$, and

$$\frac{C(R^H)}{C(R^*)} \leq \frac{n-2}{q} + 2$$

if $K \bmod q > 0$.

We can also derive the following results from the partial-load case with waiting cost when moving depots are introduced.

Theorem A.6 We have

$$C(R^*) \geq \max \{ \bar{r}_{T^*}^1, \bar{C}(T^*) \}.$$

For each $h = 1, 2, \dots, q$, if $n \bmod h = 0$, then

$$C(R^*) \leq C(R^H) \leq \frac{n}{h} \bar{r}_{T^*}^h + \Delta \left(1 - \frac{1}{h} \right) \bar{C}(T^*),$$

and if $n \bmod h > 0$, then

$$C(R^*) \leq C(R^H) \leq \frac{2n}{h} \bar{r}_{T^*}^h + \Delta \left(1 - \frac{2}{h} \right) \bar{C}(T^*),$$

and

$$\frac{C(R^H)}{C(R^*)} \leq \min_{\alpha \in \{1, 2, \dots, q\}} \mathcal{H}'(\alpha).$$

A bound independent of T^* reads

$$\frac{C(R^H)}{C(R^*)} \leq \frac{n-\Delta}{q} + \Delta$$

if $K \bmod q = 0$, and

$$\frac{C(R^H)}{C(R^*)} \leq \frac{n-2\Delta}{q} + 2\Delta$$

if $K \bmod q > 0$.

4 Computational Results

We have implemented these heuristics and extensively ran them on both randomly-generated and real-world PCB test cases. The results as documented below are representative of results obtained on multiple runs of the heuristic over test cases of varied problem complexity and dimensions (measured specifically in two dimensions, namely, the total number of components and the number of spindles present on the head).

The first set of test instances were generated using a randomized model for a board with key parameters such as number of components, component diversity, and number of spindles as inputs. Statistical analysis of the board designs for a major electronics manufacturer partner over recent years (2007-2008) were used to determine the ranges of these parameters. Machine parameters such as velocities, on the other hand, were fixed to those values found most commonly in such configurations. As a reference point, a mid-size instance with 250 board nodes, 10 spindles, 50 component types, 100 feeder slots, and 150 feeder positions yields a complete graph with 152,500 nodes and of the order of 11.6 billion edges.

The second set of test instances comprised of three real-world PCBs used by a partner electronics manufacturer. These PCBs belong to representative scales of boards typical in the industry. The smallest instance

is a board of low complexity containing 41 components belonging to one of the 10 different component types. The second instance is a board of medium complexity containing 185 components belonging to one of the 13 different types. The final instance is a board of large complexity containing 540 components belonging to one of the 61 types.

The experiments were carried out on a cluster of 52 nodes, each having two 64-bit 3.2GHz Pentium 4 Xeon processors sharing 6GB RAM with InfiniBand, running Red Hat Enterprise Linux 4 and compiled under GCC version 4.1.2. The implementation however, does not exploit parallel processors; it is sequential. We used the Concorde Lin-Kernigan TSP Solver [2] to solve TSPs and an in-house implementation of a version of the Dijkstra’s algorithm [10] to solve shortest path problems.

For typical test-case scenarios, the worst case ratios presented in this paper lie anywhere between 10 to 30 depending on the particular parameters and features of each test-case instance. In practice, the proposed algorithms tend to perform much better. It is hard to establish the ratio of the heuristics since optimal solutions or lower bounds are unknown. Instead, we compare them against the best known solution. The solutions generated by our heuristics were used as initial starting solutions to a state-of-the-art computationally intensive iterative improvement algorithm [27] based on solving integer linear programs at each iteration. They were typically found to be between 1.5 to 3 times off from the solution obtained as a result of running the advanced improvement heuristic over long periods of time (of the order of 45 minutes to 5 hours). Any ‘single-shot’ solution to such a difficult problem could be vastly improved by the application of very computationally intensive improvement heuristics. In this work, the goal is the derivation of a good starting solution to this extremely complex and challenging problem that could be further refined by various improvement and exchange heuristics available in the reserach literature, as is a commonly accepted framework for solution methodologies in the VRP and TSP world. As a testament to this, we ran the same advanced improvement algorithm with a completely randomized solution and it did not approach the solution quality of the heuristics proposed in this article even after 3 hours of crunching integer linear programming models (we unfortunately cannot compare our results to the mathematical optimum due to the intractability and immense complexity of solving even small problems exactly and thus this comparison is used herein).

In order to further gauge the quality of the heuristics, we also report the generated solution of the heuristic as being consistently about 50% better than the best solution obtained by running a genetic algorithm using genetic operators reported in literature [18] for such problems for 3 hours starting from a completely random solution pool. The running time of the heuristic averages from 2 to 10 minutes for common board instances. The heuristics are valuable in their contribution to the state of the progress towards obtaining an acceptable solution of such problems as such remain unacheivable even by advanced computational techniques without the foundation of such a strong starting solution.

The performance of the initial heuristic as compared to the best solution obtained by the iterative improvement algorithm from [27] and the genetic algorithm is depicted in Figure 14. In this figure, ‘GA’ stands for the genetic algorithm and ‘ILPBRH’ represents the iterative integer linear programming based improvement algorithm. Both algorithms have been started from the solution obtained by the heuristic presented herein, and this demonstrates the effectiveness of the advanced improvement algorithm that is compared against in lieu of a true optimum. Figure 15 shows this same algorithm being started with a randomized solution as opposed to a solution generated by the proposed heuristics.

The computational time of the heuristic grows only linearly with the total number of nodes n and it grows logarithmically and flattens out with an increasing vehicle capacity q . In Tables 1 and 2, the column ‘Soln. time’ shows the total running time of the heuristic. These trends are in line with our expectations given the nature of the heuristic. From the PCB perspective, an increasing number of components yields a near-linearly increasing network size in terms of the number of nodes, whereas an increasing number of spindles corresponds to an increasing network size for the GTSP, but simultaneously also causes a decrease in the number of routes required, which at some point becomes the dominating factor and causes the flattening effect.

5 Conclusions

In this paper, we derive worst-case bounds on several versions of the iterated-tour partitioning algorithm adapted for a superset of many important generalizations of the vehicle routing problem and its commonly

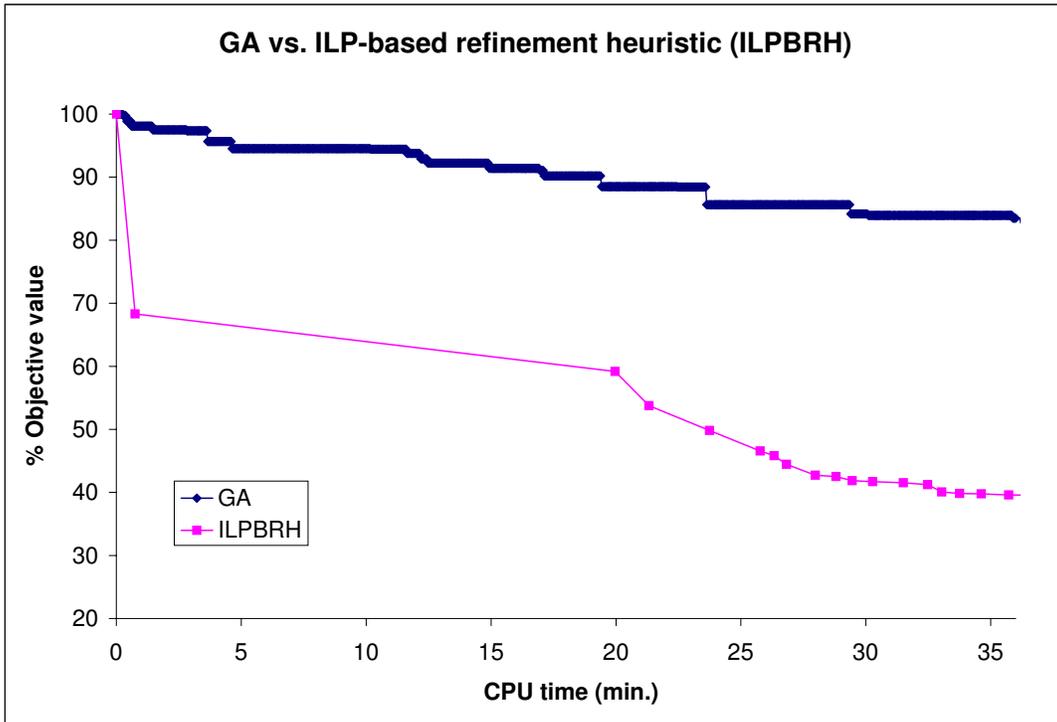


Figure 14: GA and ILPBRH starting with heuristic solution

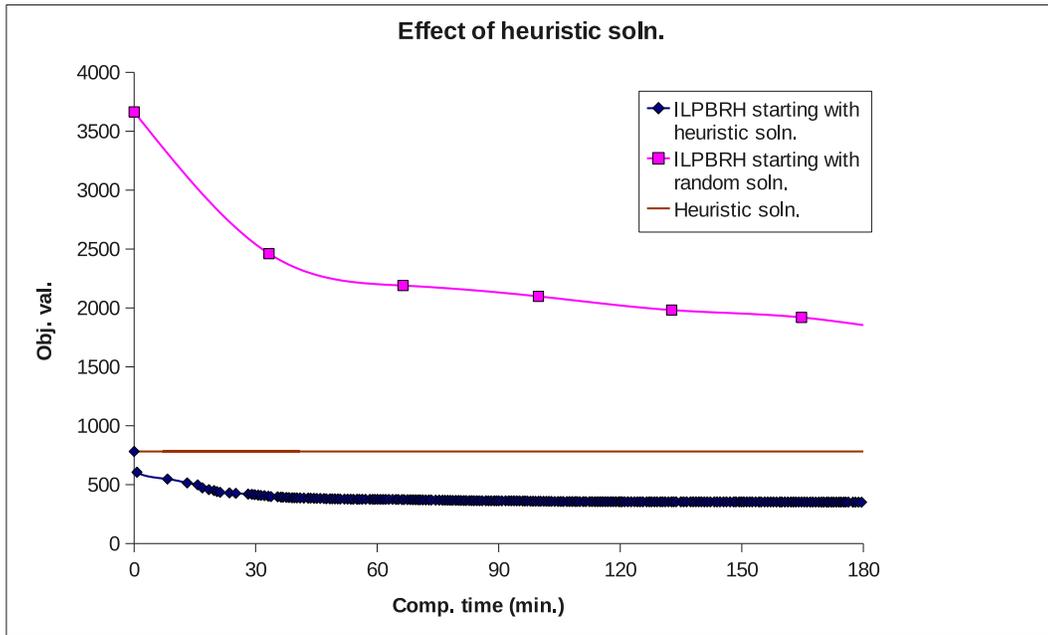


Figure 15: ILPBRH starting with random and heuristic solutions

Table 1: Effect of increasing n

n	Soln. time (mins.)
29	0.53
67	1.27
129	3.77
241	4.81
461	10.44

Table 2: Effect of increasing q

q	n	Soln. time (mins.)
6	202	4.59
12	202	14.55
18	202	45.66
24	202	93.53

found variants such as the VRP with backhauls, the multi-depot VRP, and the VRP with pick-up and delivery. These bounds have also been proved to be tight. The analyses give us an insight into the development of efficient algorithms that yield solutions that are provably within a given ratio of the optimal solution and, furthermore, are tight. The presented heuristics also have a solid practical performance by being 1.5 to 3 times worse than the best known solution.

An interesting and significant extension of this work might be to take into account the multi-commodity nature of such problems, and develop and analyze ITP heuristics under such conditions. Another interesting extension might be to allow for other constraints found in VRP scenarios, such as the time window constraints, precedence constraints, and depot stocking capacity constraints amongst others.

References

- [1] ALTINKEMER, K., AND GAVISH, B. Technical note—Heuristics for delivery problems with constant error guarantees. *Transportation Science* 24, 4 (1990), 294–297.
- [2] APPLGATE, D., BIXBY, R., CHVATAL, V., AND COOK, W. CONCORDE TSP solver. *Website* <http://www.tsp.gatech.edu/concorde.html> (2009).
- [3] BEASLEY, J. E. Route first–Cluster second methods for vehicle routing. *Omega* 11, 4 (1983), 403–408.
- [4] BERMAN, S., EDAN, Y., AND JAMSHIDI, M. Navigation of decentralized autonomous automatic guided vehicles in material handling. *IEEE Transactions on Robotics and Automation* 19, 4 (2003), 743–749.
- [5] BOMPADRE, A., DROR, M., AND ORLIN, J. B. Improved bounds for vehicle routing solutions. *Discrete Optimization* 3, 4 (2006), 299 – 316.
- [6] BRAMEL, J., AND SIMCHI-LEVI, D. A location based heuristic for general routing problems. *Operations Research* 43, 4 (1995), 649–660.
- [7] BRAMEL, J., AND SIMCHI-LEVI, D. Probabilistic analyses and practical algorithms for the vehicle routing problem with time windows. *Operations Research* 44, 3 (1996), 501–509.
- [8] CHAWLA, S. *Graph algorithms for planning and partitioning*. PhD thesis, School of Computer Science, Carnegie Mellon University, 2005.
- [9] DAGANZO, C. *Logistics systems analysis*. Springer, 2005.

- [10] DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik* 1, 1 (1959), 269–271.
- [11] FISCHETTI, M., AND TOTH, P. The generalized traveling salesman and orienteering problems. *Combinatorial Optimization* 12 (2002), 609–662.
- [12] GENDREAU, M., LAPORTE, G., AND VIGO, D. Heuristics for the traveling salesman problem with pickup and delivery. *Computers & Operations Research* 26, 7 (1999), 699–714.
- [13] GRUNOW, M., GÜNTHER, H. O., SCHLEUSENER, M., AND YILMAZ, I. O. Operations planning for collect-and-place machines in PCB assembly. *Computers & Industrial Engineering* 47, 4 (2004), 409–429.
- [14] GUTIN, G., AND PUNNEN, A. *The traveling salesman problem and its variations*. Kluwer Academic Publishers, 2002.
- [15] HAIMOVICH, M., KAN, A. H. G. R., AND STOUGIE, L. *Vehicle routing: methods and studies*. North-Holland, 1988, ch. Analysis of heuristics for vehicle routing problems, pp. 47–61.
- [16] HAIMOVICH, M., AND RINNOOY KAN, A. H. G. Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research* 10, 4 (1985), 527–542.
- [17] HO, W., JI, P., AND DEY, P. K. Optimization of PCB component placements for the collect-and-place machines. *The International Journal of Advanced Manufacturing Technology* 37, 7 (2008), 828–836.
- [18] KULAK, O., YILMAZ, I. O., AND GÜNTHER, H. O. PCB assembly scheduling for collect-and-place machines using genetic algorithms. *International Journal of Production Research* 45, 17 (2007), 3949–3969.
- [19] LAVALLE, S. M., AND HUTCHINSON, S. A. Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation* 14, 6 (1998), 912–925.
- [20] LI, C.-L., AND SIMCHI-LEVI, D. Worst-case analysis of heuristics for multi-depot capacitated vehicle routing problems. *INFORMS Journal On Computing* 2, 1 (1990), 64–73.
- [21] MOSHEIOV, G. Vehicle routing with pick-up and delivery: tour-partitioning heuristics. *Computers & Industrial Engineering* 34, 3 (1998), 669 – 684.
- [22] PARKHI, K. Printed circuit board: World outlook. Tech. Rep. 844-F2, Visant Strategies, Inc., 2007.
- [23] PSARAFTIS, H. N. Analysis of an $O(N^2)$ heuristic for the single vehicle many-to-many Euclidean dial-a-ride problem. *Transportation Research* 17, 2 (1983), 133–145.
- [24] REINER, B., AND HAHN, K. Optimized management of large-scale data sets stored on tertiary storage systems. *IEEE Distributed Systems Online* 5, 5 (2004).
- [25] ROSENBLATT, M. J., ROLL, Y., AND ZYSER, D. V. A combined optimization and simulation approach for designing automated storage/retrieval systems. *IIE Transactions* 25, 1 (1993), 40–50.
- [26] SALONEN, K., SMED, J., JOHNSSON, M., AND NEVALAINEN, O. Grouping and sequencing PCB assembly jobs with minimum feeder setups. *Robotics and Computer Integrated Manufacturing* 22, 4 (2006), 297–305.
- [27] SETH, A., KLABJAN, D., AND FERREIRA, P. M. A novel local search integer-programming-based heuristic for PCB assembly on collect-and-place machines. Tech. rep., Northwestern University, Available from <http://www.klabjan.dynresmanagement.com>, 2009.
- [28] SIPLACE-AMERICAS. Siemens automation electronic assembly systems. *Siemens’ official website* <http://ea.automation.siemens.com> (2009).
- [29] SOLOT, P., AND VAN VLIET, M. Analytical models for FMS design optimization: A survey. *International Journal of Flexible Manufacturing Systems* 6, 3 (1994), 209–233.

- [30] STEENKEN, D., VOSS, S., AND STAHLBOCK, R. Container terminal operation and operations research-a classification and literature review. *OR Spectrum* 26, 1 (2004), 3–49.
- [31] STEIN, D. M. Scheduling dial-a-ride transportation systems. *Transportation Science* 12, 3 (1978), 232–249.
- [32] TIRPAK, T. M., NELSON, P. C., AND ASMANI, A. J. Optimization of revolver head SMT machines using adaptive simulated annealing (ASA). In *Proceedings of the Twenty-Sixth IEEE/CPMT International Electronics Manufacturing Technology Symposium* (2000), pp. 214–220.
- [33] VAN DEN BERG, J. A literature survey on planning and control of warehousing systems. *IIE Transactions* 31, 8 (1999), 751–762.