

Old Dominion University

ODU Digital Commons

---

Engineering Management & Systems  
Engineering Faculty Publications

Engineering Management & Systems  
Engineering

---

2022

## A Literature Review On Combining Heuristics and Exact Algorithms in Combinatorial Optimization

Hesamoddin Tahami  
*Old Dominion University*

Hengameh Fakhravar  
*Old Dominion University*

Follow this and additional works at: [https://digitalcommons.odu.edu/emse\\_fac\\_pubs](https://digitalcommons.odu.edu/emse_fac_pubs)



Part of the [Computational Engineering Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Computation Commons](#), and the [Operations Research, Systems Engineering and Industrial Engineering Commons](#)

---

### Original Publication Citation

Tahami, H., & Fakhravar, H. (2022). A literature review on combining heuristics and exact algorithms in combinatorial optimization. *European Journal of Information Technologies and Computer Science*, 2(2), 6-12. <https://doi.org/10.24018/ejcompute.2022.2.2.50>

This Article is brought to you for free and open access by the Engineering Management & Systems Engineering at ODU Digital Commons. It has been accepted for inclusion in Engineering Management & Systems Engineering Faculty Publications by an authorized administrator of ODU Digital Commons. For more information, please contact [digitalcommons@odu.edu](mailto:digitalcommons@odu.edu).

# A Literature Review on Combining Heuristics and Exact Algorithms in Combinatorial Optimization

Hesamoddin Tahami, Hengameh Fakhrahar

## ABSTRACT

There are several approaches for solving hard optimization problems. Mathematical programming techniques such as (integer) linear programming-based methods and metaheuristic approaches are two extremely effective streams for combinatorial problems. Different research streams, more or less in isolation from one another, created these two. Only several years ago, many scholars noticed the advantages and enormous potential of building hybrids of combining mathematical programming methodologies and metaheuristics. In reality, many problems can be solved much better by exploiting synergies between these approaches than by “pure” classical algorithms. The key question is how to integrate mathematical programming methods and metaheuristics to achieve such benefits. This paper reviews existing techniques for such combinations and provides examples of using them for vehicle routing problems.

**Keywords:** Metaheuristics, optimization-based heuristics, survey, VRP.

**Published Online:** April 29, 2022

**ISSN:** 2736-5492

**DOI:** 10.24018/ejcompute.2022.2.2.50

**H. Tahami\***

Engineering Management and Systems  
Engineering, Old Dominion University,  
Virginia, USA.

(e-mail: htahami@odu.edu)

**H. Fakhrahar**

Engineering Management and Systems  
Engineering, Old Dominion University,  
Virginia, USA.

(e-mail: hfakh001@odu.edu)

*\*Corresponding Author*

## I. INTRODUCTION

Many problems arising in areas such as scheduling and production planning, location and distribution management, Internet routing, or bioinformatics are combinatorial optimization problems (COPs). COPs are fascinating because they are frequently simple to formulate yet extremely complex to solve, which is captured by the fact that many of them are NP-hard [1]. At the same time, this difficulty and their enormous practical importance have resulted in a large number of solution techniques for them. There are two algorithms for solving problems: exact and approximation algorithms. Exact algorithms are guaranteed to identify and verify an optimal solution and prove its optimality for every finite-size instance of a COP within an instance-dependent, finite run-time or show that no feasible solution exists. If optimal solutions cannot be computed quickly enough in practice, it is common to trade the guarantee of optimality for efficiency. The assurance of finding optimal solutions is sacrificed to get very good solutions by using approximate algorithms in a reasonable amount of time.

As an exact approach, the integer programming (IP) methods and stochastic local search (SLS) algorithms as an approximation approach are two solution method classes that have had substantial success [2]. IP methods rely on the characteristic of the decision variables being integers. Some well-known IP methods are branch-and-bound, branch-and-cut, branch-and-price, and dynamic programming. Exact methods for IP have the following advantages (i) proven optimal solutions can be obtained if the algorithm succeeds, (ii) valuable information on upper/lower bounds on the

optimal solution can be obtained even if the algorithm is stopped before completion (IP methods can become approximate if we define a criterion for stopping them before solving the problem), and (iii) IP methods allow to provably prune parts of the search area in which optimal solutions cannot be found. A more advantage of IP methods is that research codes such as Minto [3] or GLPK [4] or powerful, general-purpose commercial tools such as CPLEX [5] or Xpress-MP [6], [7] are available. However, despite the known successes, exact methods have a few disadvantages. First, for many problems, the size of the practically solvable instances is rather limited. Even if an application is feasible, the variance of the computation times is typically very large when applied to different instances of the same size. Second, the memory consumption of exact algorithms can be very large and lead to early abortion. Thirdly, for many COPs, the best performing algorithms are problem-specific, and they require large development times by experts in integer programming. Finally, high-performing exact algorithms for one problem are often difficult to extend if some details of the problem formulation change. The state-of-the-art for exact algorithms is that for some NP-hard problems, very large instances can be solved fast, while for other problems, even small-size instances are out of reach.

SLS is probably the most successful class of approximate algorithms. When applied to hard COPs, the local search yields high-quality solutions by iteratively applying small modifications to a solution in the hope of finding a better one. Embedded into higher-level guidance mechanisms, which are called (general-purpose) SLS methods [2] or, more commonly, metaheuristics, this method is very successful in

achieving near-optimal (maybe optimal) solutions to several difficult problems [2], [8], [9]. Examples of well-known general-purpose SLS methods (or metaheuristics) are simulated annealing, tabu search, memetic algorithms, ant colony optimization, or iterated local search [8]. Advantages of SLS algorithms are that (i) they are the best performing algorithms available for a variety of problems, (ii) they can examine a huge number of possible solutions in a short calculation time, (iii) they are often more easily adapted to slight variants of problems and are therefore more flexible, and (iv) they are typically easier to understand and implement by the common user than exact methods. However, local search-based algorithms have several disadvantages. Firstly, they cannot prove optimality and typically do not bound the quality of the solutions they return. Secondly, they typically cannot provably reduce the search space. Thirdly, they do not have well-defined stopping criteria (this is particularly true for metaheuristics). Finally, local search methods often have problems with highly constrained problems where feasible areas of the solution space are disconnected. Another problem that occurs in practice is that, with very few exceptions [10], [11], there are no efficient general-purpose local search solvers available. Hence, although one can typically develop an SLS algorithm of reasonable performance rather quickly, many applications of SLS algorithms can require considerable development and implementation efforts if very high performance is required.

It is clear by now that IP and SLS approaches have their particular advantages and disadvantages and can be seen as complementary. Therefore, it appears to be a good idea to combine these two distinct techniques into more powerful algorithms.

When considering optimization approaches that combine metaheuristics with mathematical programming techniques, the resulting hybrid system may either be of exact or heuristic nature. Exact approaches are guaranteed to yield proven optimal solutions when given enough computation time. In contrast, heuristics only aim at finding reasonably good approximate solutions, usually in a more restricted time; performance guarantees are typically not provided. Most of the existing hybrid approaches are of a heuristic nature, and mathematical programming techniques are used to boost the performance of a metaheuristic. Exploiting solutions to exactly solvable relaxations of the original problem or searching large neighborhoods utilizing mathematical programming techniques are examples of such approaches.

Also, there are several highly successful ways to exploit metaheuristic strategies for enhancing the performance of mathematical programming techniques, and often these methods retain their exactness.

The first section will continue with a structural classification of strategies for combining metaheuristics and exact optimization techniques. In the second section, we discuss the metaheuristic approaches for routing problems. The last section is devoted to a general discussion and conclusion.

## II. STRUCTURAL MODEL FOR COMBINING METAHEURISTICS WITH EXACT APPROACH

The techniques available for COPs can be divided into two main categories: exact and heuristic methods. For every

instance of a COP, exact algorithms are guaranteed to locate and show an optimal solution. On the other hand, the run-time generally increases considerably with the size of the instance, and only small or moderately sized instances can be solved to verifiable optimality in practice. In this circumstance, the only possibility is to trade optimality for run-time, yielding heuristic algorithms for larger instances. In other words, the assurance of finding optimal solutions is sacrificed for the sake of obtaining suitable solutions in a short amount of time. Two separate heterogeneous streams, each from a distinct scientific community, were successful in solving COPs:

Integer Programming (IP) is an exact approach coming from the operations research community and based on the concepts of linear programming [10]. Among the exact methods are dynamic programming, branch-and-bound (B&B), Lagrangian relaxation-based methods, and linear and integer programming-based methods, such as branch-and-cut, branch-and-price, and branch-and-cut and-price [12], [13].

Local search with various extensions and separately developed variants, in the following called metaheuristics, as a heuristic approach. Metaheuristics include, among others, simulated annealing [14], [15], tabu search [16], [17], iterated local search [14], variable neighborhood search [18], [19], and various population-based models such as evolutionary algorithms [20], scatter search [21], [22], memetic algorithms [23], [24], and various estimation of distribution algorithms [25], [26].

In [27], the authors provide a more general classification of existing approaches combining exact and metaheuristic algorithms to combinatorial optimization that combines exact and metaheuristic algorithms, dividing them into two categories:

**Collaborative Combinations:** The term "collaboration" refers to the fact that the algorithms share information but are not the same. Exact and heuristic algorithms can be run sequentially, intertwined, or in parallel.

**Integrative Combinations:** By integration, it means that one technique is a subordinate embedded component of another technique. As a result, a distinct master algorithm can be an exact or a metaheuristic algorithm, and at least one integrated solution can be either an exact or a metaheuristic algorithm.

[28], [29] present a similar classification of hybrid algorithms, further including constraint programming. The authors discern a decomposition scheme corresponding to the integrative combinations and a multiple search scheme corresponding to collaborative combinations.

In another classification, [30], [31] classifies heuristics approaches into four categories and then shows how we can use mathematical programming in each.

**Construction heuristics:** start from "scratch" and proceed through a set of steps, each of which adds a component to the solution until a complete (feasible) solution is generated. We also label such methods decomposition approaches since they effectively decompose a larger problem into a series of sequentially executed sub-problems.

**Improvement heuristics:** start with a feasible solution and iteratively execute solution improving steps until some termination condition is met.

**Relaxation-based heuristics:** It is often the case that while a problem may be very difficult, certain relaxation to that

problem may be efficiently solvable. The solution to a relaxation generates a bound on the value of a problem's optimal solution, as such relaxations are often employed in exact mathematical programming approaches. Additionally, they can often serve as a basis for effective heuristics. Two general approaches are used. In one, the solution to relaxation is modified to generate a feasible solution to the problem of interest. The prototypical approach of this type probably involves rounding the solution to a linear programming relaxation of an integer program. The second class of relaxation-based approaches uses the dual information provided by the solution to the relaxation in a subsequently executing heuristic.

Using mathematical programming algorithms to generate approximate solutions: An exact optimization algorithm terminates with an optimal solution and a proof of optimality. In many cases, a significant portion of the total solution time is spent proving that a solution found (quickly) is optimal. Another common scenario is that a lot of computing time is spent going from a "near optimal" solution to an optimal one. With this motivation, exact mathematical programming algorithms are modified to generate very well, but not necessarily optimal, solutions in many practical settings. This class of approaches is founded on the idea of solving the mathematical programming formulation in a "relaxed" manner, i.e., by relaxing some attributes of the exact solution approach that increase solution time significantly. Premature stopping a branch-and-bound algorithm rounding of the relaxed solution and heuristic variable fixing are examples of this methodology. Also, the branch-and-price/column generation-based approaches belong to this class.

The other survey on metaheuristics is the one done by [32], [33]. The categorization suggestion is different from the one adopted in [34]. The following classes will be discussed:

1. set-covering/partitioning-based approaches, which correspond to the class of branch and- price/column generation-based approaches.
2. Local branching approaches are based on the local branching scheme proposed in [35].
3. Decomposition approaches coincide with the first-class defined in [36].

### III. MATHEURISTICS FOR VEHICLE ROUTING PROBLEM: A REVIEW

Classify Metaheuristics for vehicle routing problems into three classes, which we state verbatim [37], [38]

#### A. Decomposition Approaches

Approaches to decomposition. In a decomposition technique, the problem is broken into smaller and simpler sub-problems, and each sub-problem is given its solution method. In metaheuristics, these sub-problems are solved through mathematical programming models to optimality or sub-optimality.

#### B. Improvement Heuristics

This type of metaheuristics uses mathematical programming models to improve a solution discovered via a different heuristic approach. They're popular because they can be employed with any heuristic to get a result that the

mathematical programming model attempts to improve.

#### C. Branch-and-Price/Column Generation-Based Approaches

To solve routing problems, branch-and-price algorithms have been frequently and successfully applied. These algorithms use a set partitioning formulation, in which each feasible route is assigned to a binary or integer variable (column). The solution of the linear relaxation of the formulation is taken through column generation due to the exponential number of variables. The exact approach is adjusted in branch-and-price/column generation-based metaheuristics to speed up convergence, but the guarantee of optimality is lost. For example, the column generation phase is stopped prematurely.

The following three sections are devoted to the description of these three classes.

#### D. Decomposition Approaches

Traditionally, heuristic methods, and metaheuristics, in particular, have been primal-only methods. They are usually quite effective in solving the given problem instances, and they terminate, providing the best feasible solution found during the allotted computation time. However, disregarding dual information implies some obvious drawbacks, first of all, not knowing the quality of the proposed solution, but also have found an optimal solution at the beginning of the search and having wasted CPU time ever since, having searched a big search space that could have been much reduced, or having disregarded important information that could have been very effective for constructing good solutions. Dual information is also tightly connected with the possibility of obtaining good lower bounds (referring, here and forward, to minimization problems), another element that is not a structural part of current metaheuristics. On the contrary, most mathematical programming literature dedicated to exact methods is strongly based on these elements for achieving the obtained results. There is nothing, though, that limits the effectiveness of dual/bounding procedures to exact methods. There are, in fact, wide research possibilities both in determining how to convert originally exact methods into efficient heuristics and in designing new, intrinsically heuristic techniques, which include dual information.

There are many ways in which bounds can be derived. One of the most effective of these is decomposition techniques [39]. These are techniques primarily meant to exploit the possibility of identifying a sub-problem in the problem to solve and decompose the whole problem in a master problem and a sub-problem, which communicate via dual or dual-related information. The sub-problems are handled and solved separately. Finally, a feasible solution for the original problem is obtained from the solutions to the sub-problems. In metaheuristics, one or all the sub-problems are solved through the exact solution of a mathematical programming formulation. There are three basic decomposition techniques: Lagrangean relaxation, Dantzig- Wolfe decomposition, and Bender's decomposition. These techniques' popularity derives from their effectiveness in providing efficient bounds and from the observation that many real-world problems lead themselves to a decomposition.

Unfortunately, despite their prolonged presence in the optimization literature, there is no clear-cut recipe for



determining which problems should be solved with decompositions and which are better solved by other means. Decomposition techniques are the foremost candidates for problems inherently structured as a master and different sub-problem. Still, it is at times possible to effectively decompose the formulation of a problem that does not show such structure and enjoys advantages. Examples from the literature of effective usage of decomposition techniques (mainly Lagrangean) on single-structure problems include, e.g., set covering [40], [41], set partitioning [42]-[44] and crew scheduling [45]-[48].

Vehicle routing difficulties (VRPs), inventory routing problems (IRPs), production routing problems (PRPs), and location routing problems are all examples of this (LRPs). To solve these problems related to the class of decomposition techniques, various metaheuristics have been developed. Routing problems usually entail the following two basic considerations (along with other judgments specific to the application): the clustering of customers assigned to each vehicle and the sequencing of customers in vehicle routes. This feature makes it easy to adopt a cluster first-route second decomposition method, i.e., an approach in which consumers are assigned to vehicles first. Then a choice is made on how to route the customers allotted to each vehicle.

One of the most used approaches for routing problems is the cluster first-route second approach [49], [50].

The cluster first-route second strategy divides the two main decisions that characterize routing problems, i.e., namely, assigning customers to the vehicle and sequencing the consumers visited by each route. One of the first heuristic approaches for solving the conventional VRP was cluster first-route second. In the VRP, we are provided a set of customers with demand and a fleet of vehicles with sufficient capacity. The problem is to find a set of vehicle routes that will meet these customers' needs while also ensuring that each customer is only serviced once, and that the vehicle capacity is never exceeded.

The fact that clustering of consumers may be handled through the solution of a MILP motivates a metaheuristic based on a cluster first-route second approach to solve the VRP. Instead, any heuristic available for solving the Traveling Salesman Problem can be used to manage consumer routing inside each route (TSP) [67].

The first authors who proposed a cluster first-route second metaheuristics for a routing problem, specifically for the VRP, are [51]. The seed customers are chosen heuristically in the initial step of the method, and an assignment problem is solved to optimality to allocate the other customers to the seed customers. Each seed customer represents a cluster of customers. Then, routes are generated by solving a TSP on each cluster. This approach can be used for a wide variety of routing problems. The scheme was later extended to solve the VRPTW in [52]. The author [53] proposes a decomposition approach for the VRP, similar to the one proposed in [36]. The routing problem is formulated as a capacitated concentrator location problem, guiding the algorithm (CCLP). The goal is to find seed points, calculate the cost of assigning each customer to each seed point and then solve a CCLP to find the clustering of customers. After obtaining the clusters, a TSP is solved on each cluster. The authors use the algorithm for the VRP, demonstrating that the heuristic

performs well on both problems and often outperforms previous heuristics mentioned in the literature. The same authors use a similar approach to the VRPTW [54].

### E. Improvement Heuristics

Improvement heuristics are metaheuristics that combine a heuristic with the exact solution of a MILP model to improve the solution obtained by using the heuristic. There have been several approaches to combining the heuristic technique and the solution of a MILP model. This combination can go two ways, either using MILP to improve or design metaheuristics or using metaheuristics for improving known MILP techniques, even though the first of these two directions is by far more studied.

When using MILP embedded into metaheuristics, the main possibility appears to be improving local search [40]. Local branching [60], where MILP is utilized to define a suitable neighborhood to be investigated exactly by a MILP solver, is a seminal work in this direction. Essentially, only several decision variables are left free, and the neighborhood is composed of all possible value combinations of these free variables.

The idea of an exact exploration of a possibly exponential size neighborhood is at the heart of several other approaches. Very Large Neighborhood Search (VLNS) [41] is probably one of the most well-known. This method can be applied when defining neighborhood exploration as a combinatorial optimization problem itself. In this case, it could solve it quickly in this scenario, and it becomes possible for the full exploration of exponential neighborhoods. Complementary to this last is the corridor approach [61]-[63]. A would-be large exponential neighborhood is kept of manageable size by adding an exogenous constraint to the problem formulation so that the feasible region is reduced to a "corridor" around the current solution.

Several methods build around the idea of solving MILP, the neighborhood exploration problem. They differ in the way the neighborhood is defined. For example, an unconventional way of defining it is proposed in the 'dynasearch' method [43], where the neighborhood is defined by the series of moves that can be performed at each iteration, and dynamic programming is used to find the best sequence of simple moves to use at each iteration.

However, MILP contributed to metaheuristics along two other opposite lines: improving the effectiveness of well-established metaheuristics and providing the structural basis for designing new metaheuristics. As for the first line, MILP hybrids are reported for most known metaheuristics: tabu search, variable neighborhood search, ant colony optimization, simulated annealing, genetic algorithms, scatter search, etc. Particularly appealing appear to be genetic algorithms, for which several different proposals were published, with special reference to how to optimize the crossover operator. As for the second line, the proposals are different, but they still have to settle and show how they compare to a broader range of problems other than those for which they were originally presented. One example is the so-called Forward and Backward (F&B) approach [44] which implements a memory-based look ahead strategy based on the past search history. The method iterates a partial exploration of the solution space by generating a sequence of enumerative

trees of two types, called forward and backward trees. A partial solution of the forward tree has a bound on its completion cost derived from partial solutions of the backward tree and vice-versa.

#### F. Branch and Price/Column Generation-Based Approaches

Also, branch-and-price/column generation algorithms are commonly used to solve set partitioning formulations. Branch-and-price/column generation algorithms are commonly used to solve set partitioning formulations. Branch-and-price algorithms effectively solve a wide variety of routing problems, including some of the most well-known and classic ones, such as the VRP and VRPTW. They are currently the most widely used methodology. While the branch-and-price scheme is an exact successful method, and column generation is a component, it has been used to develop high-performing and efficient heuristic algorithms. Branch-and-price/column generation-based approaches are what we call heuristic approaches. They all have one thing in common: they build heuristic solutions utilizing branch-and-price and/or column generation. However, several schemes described in the literature differ in terms of how columns are formed and/or employed to get a viable answer.

In this article, the author [45] classified this approach into four classes: restricted master heuristics, heuristic branching approaches, and relaxation-based approaches. The Restricted Master Heuristic is one of the most widely utilized branch-and-price/column generation-based algorithms. This strategy is usually used in conjunction with a branch-and-price approach. The set partitioning formulation is solved on a subset of the columns obtained by the pricing problem solution, resulting in a feasible solution. The restricted master heuristic is widely used in branch and-price approaches as it enables a quick improvement of bounds and thus a speedup of the exact solution procedure. Also, they can be used as heuristic algorithms to generate the columns. The column generation phase can be done in one of two ways: either using a heuristic that ignores the dual information provided by the restricted master problem solution or using a column generation algorithm that uses the dual information but only generates a limited number of columns. The majority of approaches fall within the first category. These systems are much easier to implement as they only require a heuristic strategy for column generation and a set partitioning model. We examine ways based on heuristic column generation first and then explain approaches based on the master problem's dual information. Heuristic branching approaches are branch-and-price algorithms in which branching is performed heuristically to prune a high number of nodes of the branch-and-bound tree and thus reach a good solution rapidly to speed up the convergence of the solution method.

In column generation approaches and branch-and-price algorithms, it is important to have fast algorithms available for repeatedly solving the pricing sub-problem, i.e., identifying a variable (column) with negative reduced costs. For many hard problems, however, this sub-problem is also hard. Fast heuristics are, therefore, sometimes used for approaching the pricing problem. It's worth noting that pricing in a column with negative reduced costs is fine, even if it's not one with the minimum reduced costs. However,

after column production, it is required to demonstrate that no additional column with negative reduced costs exists, i.e., the pricing problem must be solved precisely. Otherwise, there can be no quality assurances for the final solution of the full column generation or the branch-and-price algorithm, and they must be seen as heuristic methods only. Most heuristic approaches to pricing problems are built using relatively simple construction methods. So far, more advanced metaheuristics have been applied less frequently.

Also, almost any effective B&B approach depends on some heuristic for deriving a promising initial solution, whose objective value is used as the original upper bound. Heuristics are also generally applied to some or all sub-problems in the B&B tree, as previously mentioned, to generate new incumbent solutions and related enhanced upper bounds. Appropriate upper bounds are critical to keeping the B&B tree small. As a result, metaheuristics are frequently used for these objectives. However, the additional computational effort often does not pay off when performing a relatively expensive metaheuristic at each node of a large B&B tree in a straightforward, independent way. The metaheuristic's different calls may do more or less redundant searches in similar areas of the search space. It is therefore critical to carefully pick the B&B tree nodes for which the metaheuristic is applied, as well as the amount of effort put into each call. For example, [45] offers a chunking-based selection technique for determining whether a reactive tabu search is called at each node of the B&B tree. The chunking-based strategy measures a distance between the current node and nodes investigated by the metaheuristic to bias the selection toward distant points. According to the reported computational results, introducing the metaheuristic enhances B&B performance.

An optimal solution for a relaxation of the original problem typically shows where good or even ideal solutions might be found in the original problem's search area. As a result, solutions to relaxations are commonly used in (meta-) heuristics.

Sometimes an optimal solution to relaxation can be repaired by a problem-specific procedure to make it feasible for the original problem and use it as a promising starting point for a subsequent metaheuristic (or exact) search. Linear programming (LP) relaxation is often used for this purpose, and only a simple rounding scheme is needed. For example, [46], [66] combines interior point methods and metaheuristics to solve the multidimensional knapsack problem (MKP). In the first step, an interior point method is performed with early termination. A population of different feasible candidate solutions is formed by rounding and applying multiple different ascent heuristics. A path-relinking/scatter search is performed using this collection of solutions as the beginning population. The obtained results indicate that the proposed combination is a promising research direction

Besides initialization, optima of LP relaxations are often exploited to guide local improvement or repair infeasible candidate solutions. For example, in [47], the MKP is considered, and variables are sorted according to increasing LP values. A greedy repair mechanism considers the variables in this sequence, which removes items from the knapsack until all constraints are met. Items are considered in reverse order and included in the knapsack in a greedy

improvement approach as long as no constraint is violated.

A more direct and constrained method of exploiting the optimal solution of an LP relaxation is as follows:

Some of the decision variables with integer values in the LP-optimum are fixed, and the subsequent optimization only considers the remaining variables. Such approaches are sometimes also called core methods since the original problem is reduced and only its “hardcore” is further processed. The selection of the variables in the core is critical. Another example of exploiting the LP relaxation within metaheuristics is the hybrid tabu search algorithm [48]. Additional limits fix the total number of objects to be packed, reducing and parting the search space. By solving modified LP relaxations, bounds for these constraints can be found. Tabu search is done individually to each remaining section of the search space, beginning with a solution generated from the partial problem's LP relaxation.

The approach has further been improved in [49], [65] by additional variable fixing.

Other relaxations, in addition to the LP relaxation, are sometimes successfully used in conjunction with metaheuristics. The main approaches for putting together such combinations are similar.

The relaxation-based approaches are characterized by the fact that a feasible solution to the problem is generated from the information provided by the optimal solution of a relaxation of the master problem. Column generation is used to solve relaxation. Once the relaxed solution is obtained, a heuristic procedure is used to generate a feasible solution to the problem.

Overall, Branch-and-price/column generation-based metaheuristics are becoming more and more popular. This is due to the success of branch-and-price algorithms, which were created to solve routing problems precisely. The scientific community has amassed a vast knowledge of column generation methods, which is currently being applied to creating heuristic systems. Another benefit of branch-and-price/column generation-based approaches is that they are adaptable to various problem characteristics. Most of the methods employ a set partitioning formulation and rely on heuristic approaches to generate columns.

#### IV. CONCLUSION

We have surveyed many examples where more powerful optimization systems were constructed by combining mathematical programming techniques and metaheuristics. Many very different ways exist for such hybridizations, classified them into several major methodological categories. And also brought some examples of using them in vehicle routing problems. The most traditional approach is to use some metaheuristics to provide high-quality incumbents and bounds to a B&B-based exact method. On the other hand, quickly solved relaxations or the primal-dual relationship are often used for guiding or narrowing the search in metaheuristics. A relatively new and highly promising stream is those methods in which B&B is modified in some way to follow the spirit of local search-based metaheuristics. A frequently and successfully applied approach is a large neighborhood search through ILP techniques. We come to solution merging approaches when expanding this concept to

searching the neighborhood defined by the common and disjoint attributes of two or more parental solutions.

Furthermore, highly promising hybrid approaches are those where metaheuristics are utilized within more complex branch-and-cut and branch and-price algorithms for cut separation and column generation, respectively. As previously stated, several of the literature's approaches can be classified into multiple methodological categories we've established. Although such hybrid systems have a lot of experience, determining which algorithms and types of combinations are the most promising for a new challenge can be difficult.

#### REFERENCES

- [1] Garey MR, Johnson DS. *Computers and Intractability*, San Francisco. CA: W. H. Freeman. 1979.
- [2] Hoos HH, Stützle T. Stochastic local search: Foundations and applications. *Elsevier*; 2004.
- [3] Nemhauser GL, Savelsbergh MW, Sigismondi GC. MINTO, a mixed INTEger optimizer. *Operations Research Letters*. 1994; 15(1):47-58.
- [4] Oki E. *Linear programming and algorithms for communication networks: a practical guide to network design, control, and management*. CRC Press; 2012.
- [5] Subhaa R, Jawahar N. ILOG CPLEX OPL modelling for machine cell formation. *Int J Eng Tech*. 2013; 5: 3734-41.
- [6] Laundry R, Perregaard M, Tavares G, Tipi H, Vazacopoulos A. Solving hard mixed-integer programming problems with Xpress-MP: A MIPLIB 2003 case study. *INFORMS Journal on Computing*. 2009; 21(2): 304-13.
- [7] Toth P, Vigo D, editors. The vehicle routing problem. *Society for Industrial and Applied Mathematics*; 2002.
- [8] Glover FW, Kochenberger GA, editors. *Handbook of metaheuristics*. Springer Science & Business Media; 2006.
- [9] Michel L, See A, Hentenryck PV. Distributed constraint-based local search. In *International Conference on Principles and Practice of Constraint Programming*. Springer, Berlin, Heidelberg. 2006.
- [10] Dantzig GB. *Linear Programming and Extensions*. Princeton, New Jersey: Princeton Univ. Press. 1963.
- [11] Wolsey LA, Nemhauser GL. *Integer and combinatorial optimization*. John Wiley & Sons; 1999.
- [12] Kirkpatrick S, Gelatt Jr CD, Vecchi MP. Optimization by simulated annealing. *Science*. 1983; 220(4598): 671-80.
- [13] Glover F, Laguna M. Tabu search. In *Handbook of combinatorial optimization*. Springer, Boston, MA. 1998.
- [14] Lourenço HR, Martin OC, Stützle T. Iterated local search. In *Handbook of metaheuristics*. Springer, Boston, MA. 2003.
- [15] Hansen P, Mladenović N. An introduction to variable neighborhood search. In *Meta-heuristics*. Springer, Boston, MA. 1999.
- [16] Bäck T, Fogel DB, Michalewicz Z. Handbook of evolutionary computation. *Release*. 1997; 97(1): B1.
- [17] Glover F, Laguna M, Martí R. Fundamentals of scatter search and path relinking. *Control and Cybernetics*. 2000; 29(3): 653-84.
- [18] Moscato P, Cotta C. *A gentle introduction to memetic algorithms*. In Handbook of metaheuristics. Springer, Boston, MA. 2003.
- [19] Larrañaga P, Lozano JA, editors. *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer Science & Business Media; 2001.
- [20] Puchinger J, Raidl GR. Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In *International work-conference on the interplay between natural and artificial computation*. Springer, Berlin, Heidelberg. 2005.
- [21] Danna E, Pape CL. Two generic schemes for efficient and robust cooperative algorithms. In *Constraint and Integer Programming*. Springer, Boston, MA. 2004.
- [22] Ball MO. Heuristics based on mathematical programming. *Surveys in Operations Research and Management Science*. 2011; 16(1): 21-38.
- [23] Doerner KF, Schmid V. Survey: matheuristics for rich vehicle routing problems. In *International Workshop on Hybrid Metaheuristic*. Berlin, Heidelberg. 2010.
- [24] Fischetti M, Lodi A. Repairing MIP infeasibility through local branching. *Computers & Operations Research*. 2008; 35(5): 1436-45.
- [25] Archetti C, Speranza MG. A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*. 2014; 2(4): 223-46.



- [26] Bazaraa MS, Jarvis JJ, Sherali HD. *Linear programming and network flows*. John Wiley & Sons; 2008.
- [27] Fakhraar H. Fuzzy inventory model with receiving reparative order and considering imperfect quality items. 2020.
- [28] Caprara A, Fischetti M, Toth P. A heuristic method for the set covering problem. *Operations research*. 1999; 47(5): 730-43.
- [29] Ceria S, Nobili P, Sassano A. A Lagrangian-based heuristic for large-scale set covering problems. *Mathematical Programming*. 1998; 81(2): 215-28.
- [30] Atamtürk A, Nemhauser GL, Savelsbergh MW. A combined Lagrangian, linear programming, and implication heuristic for large-scale set partitioning problems. *Journal of Heuristics*. 1996; 1(2): 247-59.
- [31] Tahami H, Mirzazadeh A, Arshadi-khamseh A, Gholami-Qadikolaei A. A periodic review integrated inventory model for buyer's unidentified protection interval demand distribution. *Cogent Engineering*. 2016; 3(1): 1206689.
- [32] Boschetti MA, Mingozzi A, Ricciardelli S. A dual ascent procedure for the set partitioning problem. *Discrete Optimization*. 2008; 5(4): 735-47.
- [33] Van Krieken MG, Fleuren H, Peeters R. A Lagrangean relaxation based algorithm for solving set partitioning problems. 2005.
- [34] Boschetti MA, Mingozzi A, Ricciardelli S. An exact algorithm for the simplified multiple depot crew scheduling problem. *Annals of Operations Research*. 2004; 127(1): 177-201.
- [35] Tahami H, Fakhraar H. A fuzzy inventory model considering imperfect quality items with receiving reparative batch and order. *European Journal of Engineering and Technology Research*. 2020; 5(10): 1179-85.
- [36] Freling R, Huisman D, Wagelmans AP. Models and algorithms for integration of vehicle and crew scheduling. *Journal of Scheduling*. 2003; 6(1): 63-85.
- [37] Hoffman KL, Padberg M. Solving airline crew scheduling problems by branch-and-cut. *Management Science*. 1993; 39(6): 657-82.
- [38] Mingozzi A, Boschetti MA, Ricciardelli S, Bianco L. A set partitioning approach to the crew scheduling problem. *Operations Research*. 1999; 47(6): 873-88.
- [39] Tahami H, Fakhraar H. Multilevel Reorder Strategy-based Supply Chain Model. *In 5th North American Conference on Industrial Engineering and Operations Management (IEOM)*, Michigan, USA 2020.
- [40] Fisher ML, Jaikumar R. A generalized assignment heuristic for vehicle routing. *Networks*. 1981; 11(2): 109-24.
- [41] Koskosidis YA, Powell WB, Solomon MM. An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints. *Transportation science*. 1992; 26(2): 69-85.
- [42] Fakhraar H, Tahami H. Systems statistical engineering-hierarchical fuzzy constraint propagation. 2021.
- [43] Bramel J, Simchi-Levi D. A location based heuristic for general routing problems. *Operations research*. 1995; 43(4): 649-60.
- [44] Fakhraar H, Tahami H. International Co-Branding and Firms Finance Performance. arXiv preprint arXiv:2202.07128. 2022 Feb 15.
- [45] Bramel J, Simchi-Levi D. Probabilistic analyses and practical algorithms for the vehicle routing problem with time windows. *Operations Research*. 1996; 44(3): 501-9.
- [46] Fakhraar H. Quantifying uncertainty in risk assessment using fuzzy theory. arXiv preprint arXiv:2009.09334. 2020 Sep 20.
- [47] Fakhraar H. Application of Failure Modes and Effects Analysis in the Engineering Design. 2021.
- [48] Dumitrescu I, Stützle T. Usage of exact algorithms to enhance stochastic local search algorithms. *In Matheuristics*. Springer, Boston, MA. 2009.
- [49] Ouabira MM, Fakhraar H. Effective Project Management and the Role of Quality Assurance throughout the Project Life Cycle. *European Journal of Engineering and Technology Research*. 2021; 6(5).
- [50] Ahuja RK, Ergun Ö, Orlin JB, Punnen AP. A survey of very large-scale neighborhood search techniques. *Discrete Applied Mathematics*. 2002; 123(1-3): 75-102.
- [51] Sniedovich M, Viß S. The corridor method: a dynamic programming inspired metaheuristic. *Control and Cybernetics*. 2006; 35(3): 551-78.
- [52] Congram RK, Potts CN, van de Velde SL. An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem. *INFORMS Journal on Computing*. 2002; 14(1): 52-67.
- [53] Bartolini E, Mingozzi A. Algorithms for the non-bifurcated network design problem. *Journal of Heuristics*. 2009; 15(3): 259-81
- [54] Yahoodik S, Tahami H, Unverricht J, Yamani Y, Handley H, Thompson D. Blink Rate as a Measure of Driver Workload during Simulated Driving. *In Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Sage CA: Los Angeles, CA: SAGE Publications. 2020.
- [55] Woodruff DL. A chunking based selection strategy for integrating meta-heuristics with branch and bound. *In Meta-Heuristics*. Springer, Boston, MA. 1999.
- [56] Fakhraar H. Application of Failure Modes and Effects Analysis in the Engineering Design Process. arXiv preprint arXiv:2101.05444. 2021 Jan 14.
- [57] Tahami H, Fakhraar H. A fuzzy inventory model considering imperfect quality items with receiving reparative batch and order overlapping. arXiv preprint arXiv:2009.05881. 2020 Sep 12.
- [58] Woodruff DL. A chunking based selection strategy for integrating meta-heuristics with branch and bound. *In Meta-Heuristics*. Springer, Boston, MA. 1999.
- [59] Plateau A, Tachet D, Tolla P. A hybrid search combining interior point methods and metaheuristics for 0-1 programming. *International Transactions in Operational Research*. 2002; 9(6): 731-46.
- [60] Raidl GR. An improved genetic algorithm for the multiconstrained 0-1 knapsack problem. *In 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360)* 1998 May 4 (pp. 207-211). IEEE.
- [61] Vasquez M, Hao JK. A hybrid approach for the 0-1 multidimensional knapsack problem. *IJCAI*. 2001.
- [62] Vasquez M, Vimont Y. Improved results on the 0-1 multidimensional knapsack problem. *European Journal of Operational Research*. 2005; 165(1): 70-81.
- [63] Tahami H, Mirzazadeh A, Gholami-Qadikolaei A. Simultaneous control on lead time elements and ordering cost for an inflationary inventory-production model with mixture of normal distributions LTD under finite capacity. *RAIRO-Operations Research*. 2019; 53(4): 1357-84.
- [64] Bramel J, Simchi-Levi D. A location based heuristic for general routing problems. *Operations Research*. 1995; 43(4): 649-660.
- [65] Fakhraar H. Combining heuristics and Exact Algorithms: A Review. arXiv preprint arXiv:2202.02799. 2022 Feb 6.
- [66] Fakhraar H. Quantifying uncertainty in risk assessment using fuzzy theory. arXiv preprint arXiv:2009.09334. 2020 Sep 20.
- [67] Natarajan G, Ng EH, Katina PF. Systems statistical engineering-hierarchical fuzzy constraint propagation. 2021.



**H. Tahami** is a Ph.D. candidate in Engineering Management and Systems Engineering at Old Dominion University. He received his M.S. degree in Industrial Engineering. His area of research includes Supply chain optimization, Transportation, Humanitarian Logistics, and Data analysis.



**H. Fakhraar** is a Ph.D. student in Engineering Management and Systems Engineering at Old Dominion University. She received her M.S. degree in Industrial Engineering. She is working as a Graduate Assistant in the Engineering Management and Systems Engineering Department. Her research interests are Statistical Analysis, Fuzzy Methods, Systems Engineering.