Multi-thread integrative cooperative optimization for rich combinatorial problems

Teodor Gabriel Crainic
Gloria Cerasela Crisan
École des sciences de la gestion, U.Q.A.M.
Département de management et technologie
and
Centre Interuniversitaire de Recherche sur les Réseaux d’Entreprise, la Logistique et le Transport
C.P. 8888, Succursale Centre-ville, Montréal (QC), Canada H3C 3P8
(TeodorGabriel.Crainic,Cerasela.Crisan)@cirrelt.ca

Michel Gendreau
Université de Montréal
Département d’informatique et de recherche opérationnelle
and
Centre Interuniversitaire de Recherche sur les Réseaux d’Entreprise, la Logistique et le Transport
C.P. 6128, Succursale Centre-ville, Montréal (QC), Canada H3C 3J7
Michel.Gendreau@cirrelt.ca

Nadia Lahrichi
Walter Rei
École des sciences de la gestion, U.Q.A.M.
Département de management et technologie
and
Centre Interuniversitaire de Recherche sur les Réseaux d’Entreprise, la Logistique et le Transport
C.P. 8888, Succursale Centre-ville, Montréal (QC), Canada H3C 3P8
(Nadia.Lahrichi,Walter.Rei)@cirrelt.ca

Abstract

Addressing multi-attribute, “rich” combinatorial optimization problems in a comprehensive manner presents significant methodological and computational challenges. In this paper, we present an integrative multi-thread cooperative optimization framework that can simultaneously deal with multiple dimensions of a rich problem. We present the basic concepts and detail the design and operating principles of the methodology. We illustrate the framework on a rich combinatorial problem, an extended version of the vehicle routing problem with the duration and capacity constraints as well as time windows, multiple periods and multiple depots.

1 Introduction

Combinatorial optimization problems in particular, appear prominently in many theoretical and application settings. In most cases of interest, they are formally difficult and of large dimensions. A large number of methodological developments targeted these problems proposing exact, heuristic, and meta-heuristic solution methods. Parallel computing enhanced the optimization methods providing the means to accelerate the resolution process and, for meta-heuristics, to obtain higher-quality solutions for a broad range of problems [8, 6].

Many actual problems are quite complex, however, several “attributes” characterizing their feasibility and optimality structures (e.g., the many challenging characteris-
tics and complicating issues displayed by routing vehicles in real world applications [14]), which severely challenges our methodological capability to efficiently address them. The general approach when addressing such multi-attribute, rich problems is to either simplify them, or to sequentially solve a series of particular cases, where part of the overall problem is fixed or ignored, or both. It is well-known that this leads to suboptimal solutions. Moreover, one observes in many application settings, e.g., vehicle routing, wireless network design, and carrier service network design, the need to comprehensively address combinatorial problems, accounting for “all” their attributes simultaneously.

Actually, even as solution methods become more powerful, the combinatorial problems one faces grow in complexity, size and number of attributes. The currently literature does not, however, offer a satisfactory answer to this challenge in terms of methods able to efficiently address rich, multi-attribute problems and provide good solutions. Our goal is to contribute toward addressing this challenge.

We introduce a multi-thread cooperative search method, denoted Integrative Cooperative Search (ICS), for rich combinatorial optimization problems. In ICS, independent exact or meta-heuristic solution methods work on different subsets of attributes of the problem, while other algorithms combine the resulting partial solutions, and improve them. These solvers cooperate through an adaptive Global Search Coordinator, using the central-memory cooperative search paradigm. Our goal is to present and discuss the Integrative Cooperative Search concept, its structure, main building blocks, and operating principles. We illustrate these notions through an application to a multi-depot, periodic vehicle routing problem with time windows and route-length restrictions (MDPVRPTW).

This paper is organized as follows. Section 2 describes in more detail the rich combinatorial optimization problem setting we address. Section 3 outlines the main methodological ideas that lead us to the ICS concept, problem decomposition and re-combination, and cooperative search, in particular. We introduce the Integrative Cooperative Search methodology in Section 4, and discuss its application to the MDPVRPTW in Section 5. We conclude in Section 6.

2 Problem Statement

Most of the problems we are interested in addressing using the proposed approach, such as vehicle routing and network design problems, are classical combinatorial problems, which are NP-hard in their basic form. When real-world cases are considered, we face extended versions of these problems where “new” attributes have to be considered while searching for feasible solutions of high quality. Each of them compounds the complexity of the problem and complicates the resolution.

To illustrate, consider the capacitated Vehicle Routing Problem (VRP), where the objective is to construct efficient routes to deliver all customers with a fleet of vehicles with limited capacity [26]. Common generalizations consider duration constraints for the routes, time windows for the customers, frequency of delivery, different depots for vehicles, periodic deliveries, and so on. Most such settings led to the definition of generic problem definitions targeting a particular, limited subset of attributes, e.g., the capacitated VRP and the Vehicle Routing Problem with Time Windows (VRPTW). Considering all these attributes at the same time is difficult, as highlighted in a special issue on Rich VRP [16].

The most common way to deal with these multi-attribute problems is the sequential approach. According to this method, one solves the problem one dimension at a time instead of addressing it comprehensively. Sequential approaches, such as multiple-stage procedures, are widely used for Rich VRP applications [14, 15, 16], while objective separation and scalar objective aggregation appear more suitable for multi-objective applications [17]. In the latter case, two objectives are considered at a time, rarely three [2].

We believe that recent methodological progress, in parallel cooperative search methods, in particular, open the way to building methods able to address rich combinatorial optimization problems in a more comprehensive manner. To our knowledge, the only contribution addressing a rich combinatorial problem in a direct way was proposed by Crainic et al. [5] for a wireless network design problem where seven attributes were considered simultaneously. The authors proposed a parallel cooperative meta-heuristic that allowed Tabu Search (TS) solvers to work on limited subsets of attributes only, while a Genetic Algorithm (GA) amalgamated the partial solutions attained by the TS procedures. It is this decomposition-by-attribute concept that we generalize in this paper.

3 Decomposition and cooperative search

Working on subsets of attributes to address rich combinatorial optimization problems is inspired by the complementary general methods provided by the Philosophy of Science: analysis and synthesis. They are inherently intricate and follow opposite approaches, breaking down and coherently recombining solutions, respectively. This decomposition-integration approach has to be supported by an efficient optimization method. In this section, we present decomposition and integration as problem solving approaches supported by a cooperative search strategy. It should be noted that these methods are general and context free.
3.1 Problem decomposition

There is wisdom in the following citing, attributed to George Polya: “If you can’t solve a problem, then there is an easier problem you can solve: find it.” Basically, this refers to the reduction strategy, which is a heuristic method for problem solving and forms the basis for the strategy referred to earlier on of solving a series of simplified problem settings. Unfortunately, there are no analytic methods neither to devise the easier problem, nor to show how to use its result in solving the original one, which guarantee an optimal solution will be found. Another problem solving strategy is: “Look for a pattern” [22]. This translates into identifying particular structures of the problem class, which may then be used to build a solution method. Decomposition as problem-solving strategy combines these two principles, aiming for mechanisms that extract subproblems “easier” to address due to their particular structures. This approach is fundamental in many sciences, operations research and artificial intelligence, in particular.

The decomposition paradigm hides some caveats. First, decomposition very often needs to be coupled with an integration task, i.e., an efficient way to use the solutions obtained from the subproblems to build a solution to the initial problem. The latter display an optimal substructure when an optimal solution can be easily obtained from the optimal solutions of the subproblems. This is not always the case, however, and, thus, having an integration procedure does not necessarily mean that one is able to produce good quality solutions for the original problem. Second, there are supplementary computing costs when concurrently treating overlapping problems. Moreover, the dependence among the subproblems could lead to contradictory solutions; they could be integrated only after tradeoffs are considered and solved - which supposes more computing resources.

Mathematical programming offers a number of successful decomposition methods, e.g., Lagrangian relaxation and the Dantzig-Wolfe procedure, where integration is provided by the analytic framework of transformations or projections used to generate subproblems. The applicability of such exact solution methods is limited, however, for rich combinatorial optimization problems of realistic dimensions. Not only are these problems formally difficult, but the mathematical formulations amenable to the utilization of mathematical programming-based decomposition methods actually add to the complexity (in terms of number of variables and constraints, in particular).

We were inspired by such successful applications of the decomposition-integration principle, while looking for a decomposition approach free of the mathematical structure of the formulation used to model the problem considered and which would not be limited by the problem size in terms of attributes considered. We therefore use an attribute-based structural problem decomposition as our choice for further developments. Within our proposed approach, a number of independent solvers tackle, individually or in groups, the subproblems, while associated integrators concurrently and collaboratively work to re-create complete solutions and increase the global efficiency of the search.

Three major issues must be addressed relative to such a methodology and are discussed in Section 4. The first one is how to decompose the problem and define subproblems, the second is how to integrate partial solutions in order to construct and enhance solutions to the initial problem, and finally how to perform the search.

3.2 Cooperative search

The distribution of work in computer applications refers to workloads of individual processes and interactions (if any) between these individual processes. A specific type of work distribution is cooperative search. The cooperative search paradigm supposes parallel execution of several search threads using a certain level of communication. In the following, we present the taxonomy that helps us qualify the proposed ICS solution framework and the type of interactions we are using.

We use the three-dimensional taxonomy for parallel meta-heuristics from [6], which generalizes that of [11, 4, 7, 8] ([27] and [12] present classifications that proceed of the same spirit).

The first dimension examines how the global search is controlled: by a single process (1-control), or by several peer processes (p-control). The second dimension addresses how the information is exchanged, defining four levels to reflect the quantity and quality of the information exchanged and shared, as well as the additional knowledge derived from these exchanges (if any): synchronized communications decided by one controlling process or at predefined moments (Rigid Synchronized), synchronization dynamically triggered or once a pre-defined amount of exploration has been performed (Knowledge Synchronized), asynchronous communications decided by an individual process (Collegial), and asynchronous communications with creation of new information from the exchanged data (Knowledge Collegial). The third dimension describes the search differentiation in the starting point and the search strategies: Same initial Point/Population, Same search Strategy (SPSS), Same initial Point/Population, Different search Strategy (SPDS), Multiple initial Point/Population, Same search Strategy (MPSS), Multiple initial Point/Population, Different search Strategy (MPDS) - where the “point” “population” variants are used for neighborhood and population-based meta-heuristics, respectively.

The collaboration between search threads adds a supple-
mentary layer of difficulty in the model design. Historically, initial attempt considered temporal intertwining intervals, each completely dedicated either to solving or to cooperation tasks. This approach performed better than the independent search, but was later outperformed by indirect communications performed through a collection of solutions (the Central Memory model [11]) or of parts of solutions (the Adaptive Memory model [24]), or through a controlled-diffusion mechanism (the multi-level cooperative search [25]). The communication design has thus to address the exchanged information (the content), the events that trigger the exchange (the timing), the pairing of exchanging processes (the connectivity), the characteristics of communications (the mode), the purpose of the received information (the exploitation), and the further processing of the information (the scope) [9].

Figure 1. The Central Memory cooperation scheme

Figure 1 describes the Central Memory model, which forms the base of the solution framework we propose. This framework ensures a high search efficiency, as the solvers never interact directly and has been shown to displays very good performance in terms of solution quality, speed and robustness [19]. Its building blocks are:

- Algorithmic components. The individual solvers - neighborhood or population-based meta-heuristics or exact solution methods - work on the instance and build the common population of elite solutions in the central solution collection, or central memory; when a solver decides that it needs external solutions, it receives them from the central collection; a Global Search Coordinator continuously monitors the collection of solutions and builds, eventually, the information required by guidance mechanisms [20]; On request from a solver, it sends the requested solutions according to the protocols in the method design, together with guiding information (if any).

- Cooperation. Indirect exchanges through the collection of elite solutions; Communications are triggered by the internal logic of the solver deciding when to send its “good” (e.g., just improved current best) solutions to the central collection and also on the appropriate moment to ask for new solutions from the same collection.

In the next section we introduce our solution framework for rich combinatorial problems based on a p-control, Knowledge Collegial, MPDS parallel strategy with a Central Memory type of cooperation.

4 The Integrative Cooperative Search Approach

As discussed in Section 3, a broad range of decomposition and distribution strategies exist. We propose a parallel structural, attribute-based decomposition in order to handle rich, multi-attribute problems. In this framework, solvers, which in Central Memory approaches work on the full problem, now work on particular sets of attributes. We therefore denote them Partial Solvers. Consequently, a second set of solvers, denoted Integrators, cooperate to create and improve full solutions starting from the partial ones. We first present the decomposition design, then the design of the method with its algorithmic components, and finally the communication model between components.

4.1 Decomposition strategy

The Integrative Cooperative Search method uses a structural decomposition among problem attributes. This means that it investigates “facets” of the problem with dedicated solvers, and then “constructs” good complete solutions based on good partial solutions.

The idea of structural decomposition is expected to improve efficiency. On the one hand, working on selected attribute subsets is expected to provide good partial solutions rapidly. On the other hand, state-of-the-art specialized, dedicated methods may be used to address partial problems and reconstruct whole solutions.

An important issue is how to perform this decomposition. A straightforward and often used approach is to simply ignore the attributes not part of the subproblem definition. In formal terms, this is equivalent to dropping all decision variables not involved in the subproblem definition and relaxing all constraints relative to the “other” attributes. In our opinion, this approach has the drawback of creating subproblems that may have little in common with the original problem, which makes reconstructing complete solutions difficult.

We therefore propose a decomposition scheme where the attributes not part of the subproblem definition are fixed. According to this scheme, any partial solution is compatible with the other subproblems and solvers, providing the conditions for a more efficient and smooth exchange and
recombination of solutions. The initial values at which attributes are to be fixed as well as how these values are modified during the search is application specific. The main principles are detailed in the next sub-sections and illustrated in Section 5.

4.2 Design

We start our conceptual model from the Central Memory cooperation scheme, which means that the problem is concurrently tackled by several solvers and that the indirect, asynchronous communications use a dedicated memory space, the Central Memory, which collects elite solutions and may possesses guiding features [10, 20]. This concept is expanded for ICS to include Partial Solvers and Integrators as algorithmic components, as well as a broader set of tasks for the Global Search Coordinator.

Each Partial Solver is designed to investigate a subset of the problem attributes set. As in the Central Memory paradigm, Partial Solvers in the ICS model could be simple constructive methods (providing initial solutions), meta-heuristic or exact methods tailored to the subset of attributes they are dedicated to, and post-optimization methods. More than one solver may be assigned to the same set, in which case, the cooperative search model presented in the previous section applies. Partial Solvers work with complete solutions, but may modify only the values for the subset they are assigned to. The Partial Solvers dedicated to each subset of attributes construct the set of corresponding elite Partial Solutions (PS). The set of sets of PS make up a partition of the Central Memory dedicated to partial solutions. (It may be noteworthy that the actual implementation - e.g., as true common memory for all processes or as a distribution memory structure - is dependent upon the actual computer architecture used. While the implementation may impact performance, it has no impact on the ICS concept and design.)

Each Integrator constructs, and possibly improves, solutions to the initial problem, using solutions from the different PS sets. Several Integrators may work simultaneously, in which case they are also structured according to the Central Memory cooperative search model. Solutions obtained by Integrators are sent to the complete-solution partition of the Central Memory, according to the same solution quality and diversity principle (send good and diversity solutions) proper to most cooperative meta-heuristic schemes.

Figure 2 illustrates this framework, where the Central Memory now holds collections of partial solutions, each obtained from several Partial Solvers, as well as the collection of solutions to the initial problem constructed by a number of Integrators.

The Global Search Coordinator receives an extended role, mainly due to the need to guide the Partial Solvers:

- Continuously monitor the Central Memory collections and maintain the context of the global search (and, possibly, for each Partial Solver). In particular, keep for each solution a number of associated measures (i.e., simply the cost, or elaborated functions that reflect multiple characteristics) and indicators (i.e., membership to a specific class of solutions, or information on the solver which yielded it). Also build an image of the global search through statistical information, memories, on the evolution of solutions in the complete and partial memories, the contribution of solutions and their components (e.g., routes or arcs in VRP) to the evolution of the search, relative performances of Partial Solvers, etc.;

- Guide the global search, by sending “instructions” to Partial Solvers and, eventually, Integrators. This goal could be accomplished by modifying the values of the frozen attributes for a specific Partial Solver (i.e., orienting the search in another area), changing the attribute subset under investigation (i.e., changing the partition of the solution space under investigation), or replacing the solution method in a Partial Solver (or Integrator) with another one (or simply modify the method);

- Verify and enforce stopping conditions.

![Figure 2. The Integrative Cooperative Search Scheme](image)

4.3 Communication between components

Each Partial Solver decides according to its internal logic when to deposit solutions into and ask for solutions from its associated partition of the Central Memory. This process is similar to the one in the Central Memory cooperative search paradigm.

Additionally, the Global Search Coordinator decides at appropriate moments to send instructions to modify the
search trajectory of a Partial Solver. Such a decision is usually triggered by the same type of mechanisms used for diversification moves in neighborhood-based meta-heuristics or for total population renewal for evolutionary methods. Thus, for example, one may use the rate of relative improvement of the best solution in the corresponding partial solution set or in the complete solution population. Diversity measures may also be used and, of course, they may be combined to the previous ones.

The simplest form of instruction corresponds to a solution (or solution subset) intended as a starting point for the Partial Solver. (Note that the latter is not supposed to clear its search history and memories, except the short-term ones.) In more advanced mechanisms, a guiding solution context (e.g., the patterns used in [20]) could be sent together with the solution to inflect the search trajectory toward regions of the subproblem solution set that appear promising from the point of view of the global search. Instructions could also bring a change-parameter or closing command, in the latter case followed by the activation command for another Partial Solver.

With respect to Integrators, the role of the Global Search Coordinator is similar to that in the Central Memory cooperation paradigm: monitoring, responding to requests, and providing guiding information. In advanced versions where one allows the modification or replacement of solution methods, instructions similar to the ones described above also apply to Integrators.

We now illustrate these concepts through a rich VRP application.

5 A MDPVRPTW Application

The basic VRP consist of constructing optimal routes for a fleet of vehicles to serve a set of customers. Real world VRPs add a great number of attributes such as time windows, vehicle capacities, limits on the duration of routes, multiple periods, multiple depots, heterogeneous fleets, and so on. As the number of attributes grows in the same application, the choice of the solving method becomes decisive. Not only must it be adaptive, but it has to tackle all the attributes at the same time.

We are interested in this paper in a rich VRP application which can be described as a vehicle routing problem with multiple attributes. More specifically, we are interested in the multi-depot periodic vehicle routing problem with time windows (MDPVRPTW). For this problem, each customer has a determined frequency of visits in a specific planning horizon as well as a fixed time window for the deliveries. The visiting pattern has yet to be determined. Each customer can be served from one (and only one) of the depots where a fleet of vehicles is available.

Our aim in this section is to illustrate how the ICS methodology can be applied to address the MDPVRPTW and identify interesting research directions. In the following, we first present some approaches proposed in the literature for the MDPVRPTW, and then describe the ICS application. No results are currently available, but results on benchmark problems will be available for the Conference.

5.1 Previous work in MDPVRPTW decomposition

The literature on the MDPVRPTW is not very wide. Most researches have focused on either the PVRPTW or the MDVRPTW; Only a few studied the combination of these two problems. A recent survey highlights the complexity of the problem [13]. This work presents case studies and emphasizes related implementation issues, confirming the necessity for methods dedicated to this particular problem.

The most common approach for solving the MDPVRPTW is first to assign customers to a depot (generally the closest), then to assign a visiting pattern to each of them followed by solving the VRPTW. The last phase consists of an improvement phase that moves the customers from the initial depot and change their assigned visiting pattern while switching routes.

Hadjiconstantinou and Baldacci [15] define the MDPVRPTW as a multilevel combinatorial optimization problem and solve it following the four phases presented above. They use a Tabu Search algorithm to solve the VRPTW and improve the solutions obtained using a 3-opt procedure. This last phase is the only one that modifies the depot and visiting pattern assignments.

Parthanadee and Logendran [21] address the same problem using Tabu Search. All the initial assignments are built by cheapest insertion. At the improvement phase, depot and delivery pattern interchanges are used.

Another type of approach is based on generating the schedules from each depot for each day, as in [18, 28], and combining the routes generated in a second phase.

The main characteristic of all these cited works is the problem-specific way of how decomposition is made and the serial approach in addressing subproblems. In the following we present the ICS approach applied to MDPVRPTW.

5.2 Applying the ICS paradigm to MDPVRPTW

As described in Section 4, the ICS approach uses a structural decomposition based on problem attributes. To solve the MDPVRPTW problem, we decompose it onto VRPs with fewer attributes: PVRPTW and MDVRPTW. This choice of attribute decomposition is essentially based on the recent developments of effective and efficient algorithms.
to solve these problems that have exactly one less attribute than the initial problem.

While some solvers focus on the PVRPTW, the others solve the MDVRPTW. For each of these problems, the pair (client, depot) and (client, day combination), respectively, is fixed. For the PVRPTW with the (client, depot) fixed, the problem can be seen as number of depots-PVRPTWs that have to be solved, while the other reduces to number of days-MDVRPTWs. Each of these problems is handled by Partial Solvers using the Unified Tabu Search method [3], which proves to be one of the most efficient algorithms to solve these two problems. In the current implementation, two implementations with different parameter values address each subproblem and the Global Search Coordinators sends a new solutions once the improvement in the best partial solution appears to stall. In the future, we plan to add population-based algorithms, as in [23], to provide more diversity to the resolution, as well as post-optimization procedures.

The best solutions obtained are sent to the Central Memory accompanied with context information (measures, indicators, and memories). Then, in order to construct whole solutions, Integrators play their role. In the current implementation, the best partial solutions are passed to the complete solutions part of the Central Memory where the frozen attributes are allowed to be modified. This is possible due to the decomposition strategy chosen. A version of the Unified Tabu Search extended to address the MDPVRPTW is currently used to attempt to improve the solutions in the Central Memory.

Several Integrator developments are under way. The first ideas are to consider the (best) solutions of the MDVRPTW as good (client, depot) pairs, while the (best) solutions of the PVRPTW are good (client, day combination) pairs. Basic integrators combine these pairs into triplets of (client, depot, day combination) and solve a VRPTW. More advanced integrators should use informations on good routes since the solvers provide them. While all the information on routes certainly leads to infeasibility, keeping only a subset that has to be carefully chosen might produce interesting results. We propose to use methods such as path relinking and genetic algorithms to “complete” the solutions.

To test the efficiency of our approach, we will compare our algorithm in the first phase with the results obtained by the Unified Tabu Search [3], and use the test problems proposed by Cordeau [1] to create benchmark problems for MDPVRPTW. The experimentation will be conducted on dual-core AMD Opteron 248 processors, with 2.2 GHz clock speed. Procedures are coded in C++, with the communication layer written in MPI. The Single Instruction Multiple Data stream (SIMD) paradigm is used, meaning static load balancing among processors. Each processing module is assigned to a processor, and all the global data collections are managed by one separate processor.

6 Conclusion

In this paper, we present a methodological framework for tackling complex, multi-attribute problems. The strategy is to decompose the problem along subsets of attributes, to solve each resulting partial problem with dedicated methods, and to construct good solutions for the initial problem from the good solutions for the decomposed problems. All the methods work in parallel, guided by an adaptive Global Search Coordinator, using the proposed Integrative Cooperative Search paradigm. We illustrate this theoretical approach through an application to a rich VRP.

References


