Solving Multi-Objective Multi-Constraint Optimization Problems using Hybrid Ants System and Tabu Search

Hoong Chuin LAU, Min Kwang LIM, Wee Chong WAN, Hui WANG, Xiaotao WU
School of Computing, National University of Singapore
{lauhc, limmk, jwan, wanghui, wuxiaota}@comp.nus.edu.sg

Extended Abstract

1 Introduction

Many real-world optimization problems today are multi-objective multi-constraint generalizations of NP-hard problems. A classic case we study in this paper is the Inventory Routing Problem with Time Windows (IRPTW). IRPTW considers inventory costs across multiple instances of Vehicle Routing Problem with Time Windows (VRPTW). The latter is in turn extended with time-windows constraints from the Vehicle Routing Problem (VRP), which is extended with optimal fleet size objective from the single-objective Traveling Salesman Problem (TSP). While single-objective problems like TSP are solved effectively using meta-heuristics, it is not obvious how to cope with the increasing complexity systematically as the problem is compounded with additional objectives and constraints.

One common approach of handling such problems is to project these objectives onto a single weighted objective function and search for a solution that optimizes that function. Unfortunately, the correlation between these various objectives and constraints are logically weak and difficult to express generically. Typically, the optimization algorithm has no insight to which objective it is improving during the search. Consequently, much redundancy is incurred in optimizing one objective only to be undone when another objective is optimized subsequently.

In this paper, we study the effectiveness of the classical *divide-and-conquer* paradigm where sub-problems are divided along objective functions and constraints, and conquered via a hybridized meta-heuristic. The “Divide” technique involves breaking the problems into several sub-problems such that each sub-problem now contains only a single objective subject to a partial set of constraints. In addition, each sub-problem is related to another through one or more common constraints. The “Conquer” technique on the other hand, refers to a single generic scheme that is able to self-adapt through various *Derived Models* to solve different sub-problems. Each derived model represents a different degree of collaboration between two (or more) core meta-heuristics. The advantage of the derived models lies in the ability to exploit the strength and cover the weakness of the meta-heuristics under the scheme.

This paper presents an implementation of our approach on IRPTW. IRPTW is more challenging than VRPTW due to the complex relationship between the objectives and constraints. IRPTW is defined as follows, given the retailer (i.e. customer) time-varying demands over a finite planning horizon, the goal is to find a distribution and route plan that minimizes the total cost, comprising the inventory cost, backlogging cost and transportation (or routing) cost. Observe that the sub-goals of minimizing inventory and backlogging cost are conflicting, while the transportation cost becomes a non-linear component under the decomposition model of [Lau et al. 2002].

Literature survey shows that IRPTW can be solved optimally under major restrictions. [Chan et al. 1998] modeled a *single-item, constant demand* distribution system and presented worst case as well as probabilistic bounds. Unfortunately, due to the unrealistic assumption on demand, it is doubtful that any of the asymptotically optimal heuristic proposed will perform well for realistic problems with time-varying demand. [Campbell et al. 1998] considered a similar problem and proposed an integer programming model which is very computationally intensive. [Carter et al. 1996] proposed a Lagrangean heuristic to solve a single-supplier, single-warehouse. Their approach cannot guarantee feasibility (even if a solution exists), and it is sensitive to the values of several parameters where there are no good heuristics for setting them. [Lau et al. 2000] recently proposed decomposing IRPTW into two sub-problems. They then defined an interface to allow the two corresponding algorithms to collaborate in a *master-slave* fashion and provided a proof of convergence. This approach suffers 2 shortcomings: (1) inability to guarantee feasibility, when the output of the first module is infeasible for the second; and (2) the quality of solution is necessarily low, since there is no provision for sustained iterative improvement.

This work is motivated by and improves upon the work of [Lau et al. 2002], in which IRPTW is divided into VRPTW and the Dynamic Lot-sizing Problem (DLP). In addition, we also consider the decomposition of VRPTW into two single-objective sub-problems, following the scheme of [Gambardella et al. 1999]. More specifically, we present HASTS – Hybrid Ant System & Tabu Search – an adaptable scheme that hybridize the ants system and tabu search. The model exploits two similar algorithms with a key difference – one uses a preference list or better known as the pheromone trails (ants system), while the other utilizes a tabu list (tabu search). Section 2 presents details on the various derived models of HASTS. In Section 3, we illustrate how IRPTW can be solved using HASTS. Section 4 provides experimental results on the IRPTW's benchmark problems.

2. Methodology

2.1 Decomposition Approach (Divide technique)

The decomposition approach is designed to deal with optimization problems with multiple objectives and constraints. Suppose we have a problem with a multi-objective function $G(x) = \sum g_i(x)$ where g_i is a single distinct sub-objective function. We then define each sub-problem to have a sub-objective subject to a subset of constraints. A necessary condition we impose is that each sub-problem must contain at least one common constraint with those of another sub-problem so as to "bind" them together during optimization. This condition is extremely crucial as it ensures that we do not "over-optimize" a single sub-problem at the expense of another sub-problem. Formally, our decomposition technique is defined as follows.

Given a *n*-objective problem Q

$$\text{Maximize} \quad G(x) = \sum g_i(x) \quad \text{s.t.} \quad C = \{c_1, c_2, \ldots, c_m\}$$

we reformulate Q as *n* sub-problems, where the i^{th} sub-problem is defined by:

$$\text{Maximize} \quad g_i(x) \quad \text{s.t} \quad C_i \subseteq C$$

such that $C_1 \cup C_2 \cup \ldots \cup C_n = C$ and $C_i \cup C_{i+1} \neq \Phi$.

2.2 Derived Models (Conquer technique)

Having divided the problem into sub-problems, the next step is to select the most suitable Conquer technique to solve each sub-problem. In this paper, we present HASTS - Hybrid Ants System (AS) and Tabu Search (TS), a hybrid model that contains 4 derived models, which will be explained in the following sub-sections.

The standard AS builds a complete solution with each ant and the density of the pheromone trails reflects the preference of the solution structure. The pheromone trails provides information sharing and intelligence in which the quality of the solution can be optimized. As it does not require an initial solution, AS can be viewed as an excellent construction heuristic. Being a meta-heuristic, it is also not limited to a single type of problem and offers solutions of high quality. Hence, we adopt AS as a component for our hybrid model.

On the other hand, the standard TS incorporates both an adaptive memory and a responsive exploration. The adaptive memory allows TS to reduce solution cycling, and is capable of making radically changes based on past history. Responsive exploration allows TS to apply intensification and diversification strategies adaptively. However, TS is not without any weaknesses. Its effectiveness hinges on the neighborhood structure and tabu list. Hence, TS will be trapped in a poor local optimal if it lacks an effectual neighborhood and tabu list.

We deem AS and TS to be a good combination as the two meta-heuristics are very different and complementary in nature. HASTS hybridizes the two meta-heuristics to form 4 derived models, each adjusting the relative importance of AS and TS to cater to the needs of different sub-problems. The 4 derived models are *Empowered Ants* (HASTS-EA), *Improved*

Exploitation (HASTS-IE), *Enhanced Diversification* (HASTS-ED**)** and *Collaborative Coalition (*HASTS-CC), as shown in Figure 2.1.

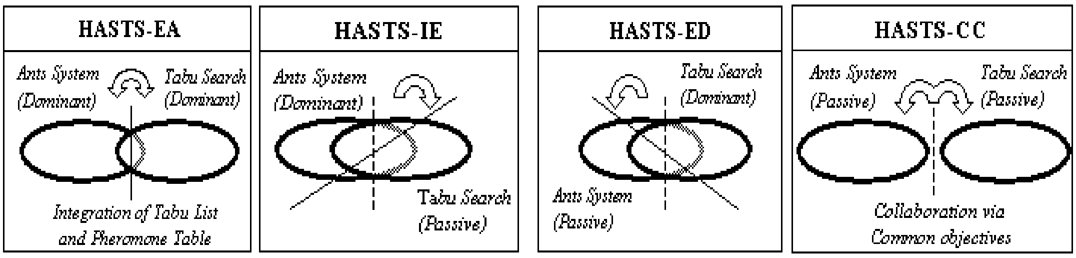

Figure 2.1: The Four Derived Models of HASTS.

2.2.1 HASTS-EA *(Empowered Ants)*

This derived model arises from the observation that while the AS reaches near optimal solutions, it suffers from a tendency of solution cycling in the near optimum region due to their emphasis on the strong pheromone trails. By empowering the ants with memory, it reduces the chances of reconstructing the same solution. An analogy can be drawn where each ant becomes more intelligent to find a better trail by *not* following false tracks laid by previous ants. TS uses a tabu list to reduce cycling on the same set of solutions. While the AS optimizes the solution based on its pheromone trails as a "preference" memory, solution cycling is reduced via the tabu list. Furthermore, TS can be applied to modify the solutions radically, hence encouraging exploration that helps to escape from local optimality. In our implementation, the AS is modified to include a tabu list, which records the solution made by each ant in a single iteration. Subsequently, each ant in the iteration would check if the next move is tabu-ed. If it is, the move will be dropped and a new move will be generated. The tabu list is reset at the end of the iteration. A pseudo-code of HASTS-EA is shown in the Appendix.

2.2.2 HASTS-IE *(Improved Exploitation)*

In this model, TS is embedded in the AS to conduct *intensification* search on the best solution. A similar design has been employed in [Stutzle and Dorigo 1999] to produce good solutions for TSP. This model offers two advantages. First, by updating the pheromone trail only after intensifying the best solution, we increase the probability of finding a better solution by subsequent ants. Second, due to the probabilistic guided nature of AS, this narrows the chances of reaching an optimal solution if it happens to be radically different from local optimum. For example, it is well known that for TSP, the AS may take a long time before it reaches optimality, due to the presence of "crossings" in the tour. With the help of TS, such crossings can be eliminated easily by swap moves such as 2-opt.

2.2.3 HASTS-ED *(Enhanced Diversification)*

In this model, AS acts as a diversifier for TS. As TS suffers from local optimality, a diversification strategy is to apply another meta-heuristic as a diversifier (e.g. [Li and Lim 2001]). HASTS-ED uses an AS diversifier with following rationale. First, the probabilistic nature of the AS gives a higher chance of successfully diversifying from the local optimum. Second, the diversifier should make a radical move from the current solution so as to explore new regions. Although a random restart is a good strategy, the new starting solution is often poor. AS provides a remedy to this by reconstructing quality solutions.

2.2.4 HASTS-CC *(Collaborative Coalition)*

This final model proposes a collaborative coalition between the AS and TS. This model offers the least coupling between the two meta-heuristics but allows great flexibility in the solution approach. One configuration of HASTS-CC is to espouse the two-phase approach as advocated by [Schulze and Fahle 1997]. This approach consists of a construction phase follow by a local improvement phase. The AS works extremely well for the construction phase as it could be used independently to obtain quality solutions. Being an optimization heuristic, TS fit naturally into the second phase of the approach. Such collaboration exploits the natural heritage of each meta-heuristic.

3. Solving IRPTW with HASTS

In this section, we apply our Divide-and-Conquer methodology to solve IRPTW.

IRPTW has three sub-objectives: (1) number of vehicles used; (2) total distance traveled; and (3) inventory and backlog costs. It has the following constraints: (1) customer time windows; (2) vehicle capacity; and (3) customer holding capacities. In [Lau et al. 2002], IRPTW is decomposed into 2 sub-problems: VRPTW and DLP (Dynamic Lotsizing Problem). Here, we divide IRPTW into 3 single-objective sub-problems VRPTW1, VRPTW2 and DLP, illustrated in Figure 3.1.

We can observe that it is the vehicle capacity constraint that binds these three sub-problems together. [Lau et al. 2002] proposed solving DLP by TS with an underlying *Minimum Cost Flow* (*MCF*) sub-problem. Following their model, we define DLP problem with *two* constraints (vehicle capacity and customer holding capacity), since MCF is able to solve the problem optimally when the route plan is fixed.

Sub-problem 1: We can reformulate this objective to its dual model and writing it as maximizing the customers served in given a set of vehicles, and reduce the required vehicles each time we find a solution that serves all the customers. We apply the ***HASTS-EA*** derived model for this sub-problem. Initially *m* vehicles are obtained by applying a greedy heuristic to

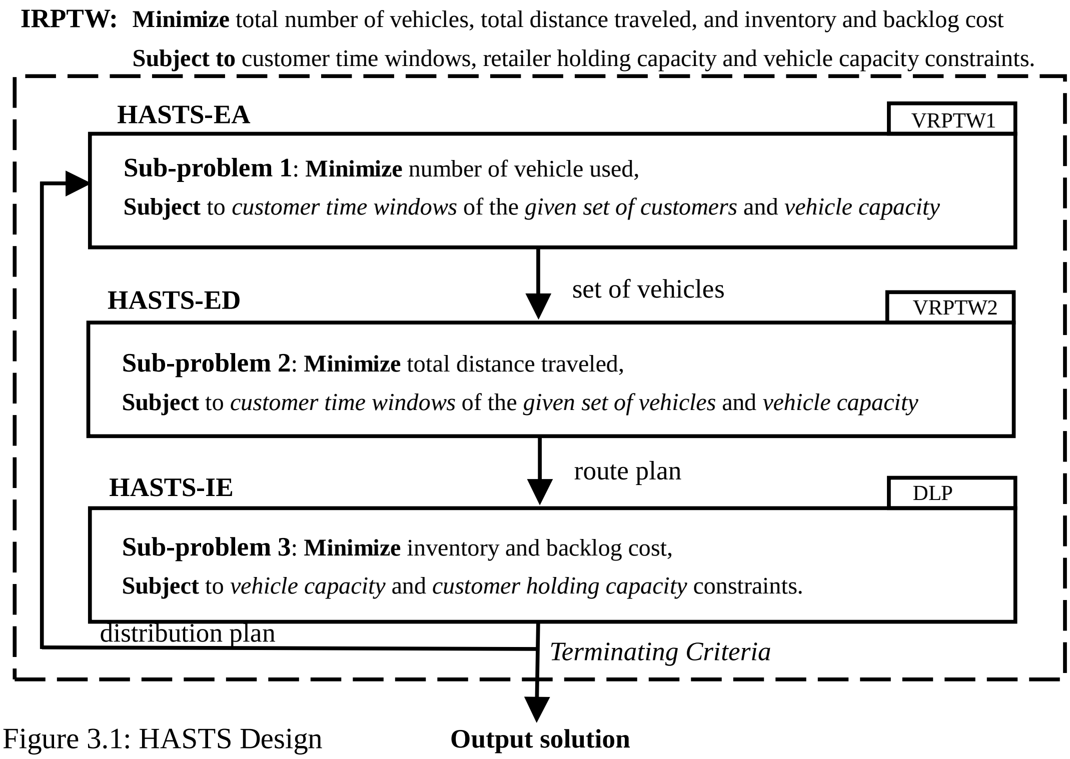

Figure 3.1: HASTS Design

serve all customers. The algorithm then reduces the value of m by 1 and seeks to construct a feasible solution that services all the customers. Once a feasible solution is found, the number of vehicles is reduced to the best-found number of vehicles and the process is repeated for a new feasible solution. This sub-problem requires search so as to find a configuration where the customers can fit into the pre-set vehicles. HASTS-EA performs well since the tabu list assists each ant in an iteration to construct a radically different solution. Although other derived models can also be used, they lack the intensified exploration that HASTS-EA provides.

Sub-problem 2: As we have already optimized on the number vehicles, this sub-problem will have a tighter solution space. HASTS-EA is not very effective in such situation because of the difficulties involved in constructing different feasible solutions on an allowed number of vehicles, due to the nature of the AS. Hence, we employ another derived model, ***HASTS-ED***, to minimize the total distance on a fixed set of vehicles. HASTS-ED uses TS as the core heuristic with AS acting as the diversifier. TS is effective in solving this sub-problem as it optimizes the route distance rather than reconstructs the solutions. When TS meets a local optimum, it randomly selects some of the routes to be reconstructed by AS. AS then assists TS by radically re-configuring the selected partial routes. The output is a route plan which is the input to Sub-problem 3.

Sub-problem 3: We adopt ***HASTS-IE*** to minimize the inventory and backlog costs. In order to reduce inventory or backlog, more frequent deliveries have to be made, hence increasing the transportation cost. Hence, the goal here is to minimize the number of customers served each day *without* increasing the total cost. Our goal is to delete retailers from routes in a manner that does not incur additional cost. HASTS-IE uses the AS to construct different solutions. It then uses TS to improve its exploitation to reduce missing elite solutions. The TS uses the standard add, delete and swap moves that attempts to improve the solution quality found by the AS. The output is a distribution plan that induces the set of customers to be served for Sub-problem 1.

4. Experiment Results

We conducted two sets of experiments, following the test instance generation strategy of [Lau et al. 2002]. The planning period is 10 days. The vehicle capacity, locations and time-windows of the customers and depot are those specified in the Solomon instances. The demand d_{it} of customer i for day t (t=1, …, 10) is equal to the demand d_i of the Solomon instance, by partitioning the value 10*d_i into 10 parts, i.e. d_{i1}, d_{i2},…,$d_{i,10}$ randomly such that d_{it} is within the range [0.5*d_i, 1.5*d_j]. The capacities of consumers and warehouse are the vehicle capacity and infinity respectively. As for cost coefficients, the inventory cost and backlog cost for each customer are 1 and 2 respectively. The transportation cost of each route is 10 times its total distance. Table 1 shows the results of C2 series and R2 series test cases when compared with [Lau et al. 2002]. The columns ILS+VRP and TS+VRP denote the results achieved authors. In Table 2, we present our results for RC2 series and R2 series cases, which have no prior results. For each test case, the average computation time is 30 minutes. Notice that although the computation time is higher than those of [Lau et al. 2002], the quality of the solutions is much improved.

Table 1: Performance comparison with [Lau et al. 2002]

Cases	ILS+VRP	TS+VRP	HASTS	R201	111330	116893	59893
C201	113263	112821	54905	R202	116982	114717	55665
C202	117483	124312	53404	R203	110215	115070	53371
C203	131920	122055	53620	R204	114118	114118	50227
C204	136384	142300	54778	R205	122333	123009	54511
C205	116147	109248	51907	R206	120928	123251	52340
C206	123978	127876	50507	R207	115438	115438	50299
C207	122204	117735	51453	R208	120011	117255	50090
C208	124110	125667	52501	R209	116840	120725	51690

Table 2: Experiment Results

Cases	HASTS
R210	53659
R211	50371
RC201	68244
RC202	67877
RC203	62453
RC204	58698
RC205	67728
RC206	66798
RC207	62596
RC208	58733

4. Conclusion

In this paper, we propose a scheme in handling multi-objective multi–constraint problems. After decomposing the problem into multiple sub-problems with single distinct objectives, we choose the most suitable derived HASTS model to solve each sub-problem. The derived models are illustrated through solving the IRPTW. On the Solomon problem set, HASTS produces good solutions within reasonable computing times.

References:

[Campbell et al. 1998] A. Campbell, L. Clarke, A. J. Kleywegt, and M. W. P. Savelsbergh, The Inventory Routing Problem, in T.G. Crainic, and G. Laporte, (eds), Fleet Management and Logistics, Kluwer Academic Publishers, pp. 95-113, 1998.

[Carter et al. 1996] M. W. Carter, J. M. Farvolden, G. Laporte, J. Xu, Solving an Integrated Logistics Problem Arising in Grocery Distribution, INFOR, Vol 34:4, pp. 290-306, 1996.

[Chan et al. 1998] L. M. Chan, A. Federgruen and D. Simchi-Levi, Probabilistic Analysis and Practical Algorithms for Inventory-Routing Models, Ops Res, Vol. 46:1. pp. 96-106, 1998.

[Gambardella et al. 1999] L. M. Gambardella, E. Taillard, and G. Agazzi. MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization, pp. 63-76. McGraw Hill, 1999.

[Lau et al. 2000] H. C. Lau, A. Lim, and Q. Z. Liu. Solving a Supply Chain Optimization Problem Collaboratively. Proc. 17th National Conf. on Artificial Intelligence (AAAI), 780-785, 2000

[Lau et al. 2002] H. C. Lau, H. Ono, and Q. Z. Liu. Integrating Local Search and Network Flow to Solve the Inventory Routing Problem. Proc. 19th National Conf. on Artificial Intelligence (AAAI), 9-14, 2002.

[Li and Lim 2001] H. Li and A. Lim. A Metaheuristic for the Pickup and Delivery Problem with Time Windows. 13th IEEE Int'l Conf on Tools with Artificial Intelligence (ICTAI), 2001

[Schulze and Fahle 1997] J. Schulze, and T. Fahle, A Parallel Algorithm for the Vehicle Routing Problem with Time Windows Constraints, 1997.

[Stutzle and Dorigo 1999] T. Stutzle and M. Dorigo, AS Algorithms for the Traveling Salesman Problem. In Evolutionary Algorithms in Engineering and Computer Science, pp. 163-183, Wiley, 1999.

Appendix. Pseudo codes of HASTS-EA

```
Procedure: HASTS – EA ()
    While (termination-criterion-not-satisfied)
        While (Max_Ant_Not_Reached)
            Ants_generation_and_activity
            Pheromone_Evaporation
            Reset_Tabu_List
            Daemon_actions
        end Schedule_activities
    end While
```

```
Procedure: Ants_generation_and_activity ()
    While (available_resources)
        Schedule_creation_of_new_ant
        New_Solution = New_active_ant
        update_Tabu_List (New_Solution)
    end While
```

```
Procedure: New_active_ant ()
    Initialize_ant;
    M = read_Pheromone Trail
    T = read_Tabu_List
    While (current_state != target_state)
        A = read_local_ant_routing_table
        P = compute_transitional_probabilities (A, M)
        For each Next_state do
        Next_state = apply_ant_decision_policy(P)
        While (check_Tabu_List (Next_state) == non-tabued)
        Move_to_next_state (next_state)
        If (online_step-by-step_pheromone_update)
            Deposit pheromone
            Update M
        end If
    end While
    If (online_delayed_pheromone_update)
        For each visited_arc do
            Deposit pheromone
            Update M
        end
    end If
```