Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 20 (2013) 46 – 51

Procedia Computer Science

Complex Adaptive Systems, Publication 3
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2013- Baltimore, MD

Employing Learning to Improve the Performance of Meta-RaPS

Fatemah Al-Duoli[a], Ghaith Rabadi[a,b]*

[a]*Department of Engineering Management and Systems Engineering, Old Dominion University, Norfolk, Virginia 23529, USA*
[b]*Visiting Associate Professor, Mechanical and Industrial Engineering Department, Qatar University, Qatar*

Abstract

In their search for satisfactory solutions to complex combinatorial problems, metaheuristics methods are expected to intelligently explore the solution space. Various forms of memory have been used to achieve this goal and improve the performance of metaheuristics, which warranted the development of the Adaptive Memory Programming (AMP) framework [1]. This paper follows this framework by integrating Machine Learning (ML) concepts into metaheuristics as a way to guide metaheuristics while searching for solutions. The target metaheuristic method is Meta-heuristic for Randomized Priority Search (Meta-RaPS). Similar to most metaheuristics, Meta-RaPS consists of construction and improvement phases. Randomness coupled with a greedy heuristic are typically employed in the construction phase. While a local search heuristic is used in the improvement phase. This paper proposes adding a new learning phase, in which a ML method will be integrated. An Inductive Decision Tree (IDT) will be incorporated into the learning phase in an effort to learn from the information collected during the construction and improvement phases. The proposed approach will be demonstrated using instances for the Capacitated Vehicle Routing Problem (CVRP).

Selection and peer-review under responsibility of Missouri University of Science and Technology
Keywords: Vehicle Routing Problem; Metaheuristics; Meta-RaPS; Supervised Learning

1. Introduction

Metaheuristic are approximate optimization methods that define search strategies and use heuristics to solve large combinatorial problems in a reasonable time [2 – old 16]. Utilizing memory is an important feature of many metaheuristic methods [1]. With memory comes the possibility of learning and building strategies that aim at finding better solutions more efficiently. The first metaheuristic method to explicitly use memory is Tabu Search [3]. However, Taillard et al (2001) [1] noted that the concept of memory can be found in many metaheuristic methods including the population in Genetic Algorithms and the pheromone trail in Ant Colony Optimization. The various forms of memory used in metaheuristics prompted Taillard et al. [1] to introduce the Adaptive Memory Programming (AMP) framework, which formalized the incorporation of memory in metaheuristics. Memory allows a variety of information to be collected during the process of solving a combinatorial problem. Converting the memory into knowledge via a learning technique helps improve the performance of metaheuristics. Integrating

* Corresponding author. Tel.: +0-757-683-4918; fax: +0-757-683-5640.
E-mail address: grabadi@odu.edu.

Selection and peer-review under responsibility of Missouri University of Science and Technology
doi:10.1016/j.procs.2013.09.237

Machine Learning (ML) algorithms into metaheuristics is one of the approaches used to make use of the memory and improve the performance of metaheuristics.

ML algorithms can be categorized according to the training samples used in the learning process. A training sample consists of data instances and/or labels depending on the learning approach, where a training instance is a set of attributes and a label is the desired prediction of that instance (or a class that the instance belongs to) [4]. Unsupervised algorithms accept unlabeled training set. Semi-supervised learning algorithms accept mixed (labeled and unlabeled) instances. Supervised learning algorithms accept labeled data instances[5]. The labeled training samples in supervised learning algorithms represent past experiences. These experiences help a supervised algorithm in making future decisions when faced with new data.

Meta-RaPS (Meta-heuristic for Randomized Priority Search) is one of the latest metaheuristics that fit under the AMP framework. This was demonstrated by Lan and DePuy [6], who incorporated memory into the memory-less Meta-RaPS baseline. Mirza [7] and Arin and Rabadi [8] have also incorporated memory and learning via integrating ML algorithms. Their efforts focused on incorporating unsupervised and semi-supervised ML algorithms. In this paper, the proposed concept aims at incorporating a supervised ML algorithm, namely *Inductive Decision Trees (IDT's)*, into Meta-RaPS and apply it to the Capacitated Vehicle Routing Problem (CVRP).

The CVRP is a variation of the Vehicle Routing Problem (VRP), which is a combinatorial problem that was introduced in the 1960s [9] for discovering optimized routes from a central depot to several customers. The VRP and its variants have been studied and solved using both exact and approximate algorithms [10]. The VRP is considered an NP-hard problem with no exact algorithms for instances that have >75 customers [11].

2. Meta-RaPS

Meta-RaPS is a metaheuristic method initially introduced as a modified form of COMSOAL heuristic (Computer Method of Sequencing Operations for Assembly Lines) [12]. Moraga et. al [14] formally defined Meta-RaPS as a "generic, high-level search procedures that introduce randomness to a construction heuristic as a device to avoid getting trapped at a local optimal solution".

Similar to other metaheuristics, Meta-RaPS divides the process of solving a problem into a construction phase and an improvement phase. Throughout these phases, Meta-RaPS uses four parameters: number of iterations (I), priority percentage (p%), restriction percentage (r%), and improvement percentage (i%).

In every iteration, Meta-RaPS solves a combinatorial problem by incrementally building a solution in the construction phase and possibly improving it in the improvement phase. During the construction phase, Meta-RaPS employs a greedy heuristic to identify a set of feasible next moves. The set of feasible next moves are prioritized. The move with the best priority is selected p% of the time. During the remaining times, (100 - p%), the next feasible move is selected randomly from a restricted list and added to the solution. The restricted list contains a set of next moves with priority values that are within the best r% of the best priority move. The construction process is repeated until a solution is complete. Subsequently, the constructed solution may be improved if its value is within i% of the best constructed solution to avoid wasting time on a very inferior solution. A local search heuristic is typically employed in an attempt to improve the constructed solution. The best solution produced from all iterations is reported as the output of Meta-RaPS.

Since its introduction, Meta-RaPS has been successfully used to solve several combinatorial problems including the Resource Constrained Project Scheduling Problem [12], the Vehicle Routing Problem [15], the Traveling Salesman Problem [2], the 0-1 Multidimensional Knapsack Problem [16], the Unrelated Parallel Machine Scheduling Problem with Setup Times [17], the Early/Tardy Single Machine Scheduling Problem [18], the Spatial Scheduling Problem with Release Times [19], and the Aerial Refueling Scheduling Problem with Due Data to Deadline Windows and Release Time [20].

3. Inductive Decision Tree (IDT)

Supervised machine learning can be achieved using Inductive Decision Trees (IDT's). The trees are logical trees that generate rules to classify data. The IDT's are constructed using a labeled training sample. Learning is achieved by using the rules (generated by the tree) in classifying new data. Several algorithms are available to construct IDT's. The algorithm used here is the ID3 algorithm introduced by Quinlan [21]. ID3 constructs an IDT recursively

by dividing the labeled training sample based on the values of an attribute with the most information gain. The algorithm uses the concept of information entropy, which measures the ambiguity found when using an attribute to classify instances [22]. Thus, the best attribute is an attribute with least entropy, which results in the most information gain.

Figure 1 shows the pseudo code for constructing a tree using ID3 algorithm [23]. For a given training sample, let the instances belong to one of two classes: positive and negative. Additionally, let the set of attributes used to describe an instance is known as A. The individual attribute is indicated as a_i where i can be 1 to $|A|$. Each attribute a_i can have a set of possible values $|V_i|$. The individual attribute value is indicated as v_i. For each node, let p denote the number of positive instances; let n denote the number of negative instances; let p_{ij} denote the number of positive instances with value v_{ij} of attribute a_i; let n_{ij} denote the number of negative instances with value v_{ij} of attribute a_i. The entropy is calculated using equation (1)

$$E(a_i) = \sum_{j=1}^{|V_i|} \frac{p_{ij} + n_{ij}}{p + n} I(p_{ij}, n_{ij}) \quad (1)$$

where

$$I(p_{ij}, n_{ij}) = -\frac{p_{ij}}{p_{ij} + n_{ij}} Log \frac{p_{ij}}{p_{ij} + n_{ij}} - \frac{n_{ij}}{p_{ij} + n_{ij}} Log \frac{n_{ij}}{p_{ij} + n_{ij}} \quad (2)$$

1. If all instances belong to one class, then the decision tree is a single node representing the name of the class
2. Otherwise,
    a. Find the best attribute with the least E value
    b. Divide instance based on the values of the best attribute
    c. Use subset of instances (b) to calculate the new best attribute

Fig. 1. Tree construction logic using ID3

ID3 starts with evaluating the attributes in the training sample. An attribute with the highest information gain (thus, best divides the training sample) is selected to be the root node of the tree. The branches of each node represent all possible values of a feature. As the tree branches out, a leaf (end) node is reached. Leaf nodes represent the outcomes of the classification, which are the classes to which instances belong.

4. Proposed Concept

The goal of this effort is to continue improving the effectiveness of Meta-RaPS by adding a learning phase that employs a supervised ML algorithm. The following sub-sections describe the proposed concept and demonstrate it by applying it to the CVRP.

4.1. The Capacitated Vehicle Routing Problem (CVRP)

The CVRP describes the layout of a system with a central depot and various destinations or customers. The goal is to find optimized routes that minimize the total distance traveled while satisfying the vehicle/truck capacity constraints. The input for the CVRP consists of the coordinates of each customer to be served. The coordinates are used to calculate the distances between the depot and each customer. Therefore, the default number of routes in a CVRP is equal to the number of customers in the problem, where each route can be described as [*0, i, 0*], where the depot is usually located at vertex 0 and *i* represents customer *i*. The symmetrical CVRP will be used to illustrate the proposed concept.

4.2. Construction Phase

The construction phase uses a greedy heuristic to incrementally build a solution. The heuristic selected is the Clarke and Wright (C&W) savings algorithm [24]. This algorithm solves the CVRP by merging routes in an effort

to maximize the savings by reducing the total distance traveled. The concept of savings is achieved in the following equation:

$$S_{ij} = d_{0i} + d_{0j} - d_{ij} \geq 0 \qquad \forall arc(i, j) \tag{3}$$

The distance between two destinations is referred to as d. Thus, the distance between customer i and the depot is d_{0i}, the distance between customers i and j is d_{ij}. S_{ij} represents the distance saved by directly connecting customers i and j. To use the C&W algorithm to solve a CVRP instance, the distances between each customer must be calculated. The distances can be saved in a distance matrix. The distances are then used to create a savings matrix, where equation (3) is used to calculate the savings if an arc were to connect two customers directly. The savings matrix is then sorted in a descending order, where the arc with the most savings is first. The outcome of sorting will be referred to as the sorted savings list.

Every arc in the sorted savings list represents a possibility to merge two CVRP routes by connecting two customers together. However, to merge two routes using an arc from the sorted savings list, the arc is examined against three C&W conditions. The first condition checks if the customers in the arc belong to two separate existing routes. The second condition verifies that the truck capacity is not exceeded if two routes are merged using the arc under consideration. The last condition checks if the both customers in the arc are either first or last in their existing routes. After evaluating every arc in the sorted savings list, several of the initial routes are expected to be merged.

When using the C&W heuristic in the construction phase of Meta-RaPS, the initial steps (forming the distance matrix, forming the savings matrix, and creating the sorted savings list) are performed first. The sorted savings list represents priorities. Thus, the construction phase will loop over every arc in the sorted savings list. If a random value $\leq p\%$, then use the first arc in the list. Else, create a restricted list using the $r\%$ and randomly select an arc. The selected arc is then evaluated against C&W conditions. If the arc passes all three conditions, then two routes are merged. This process is repeated until all the arcs have been evaluated. The total distance of the final routes is then calculated.

4.3. Improvement Phase

If the total calculated distance of the constructed solution is within $i\%$ of the best constructed solution found so far, then apply a local search heuristic. The local search heuristic chosen here is pair-wise exchange [25]. The heuristic takes every route in the constructed solution, exchanges each adjacent pairs of nodes, and recalculates the distance. If the calculated distance is shorter, then improvement is attained and the solution is kept.

4.4. Learning Phase

The learning approach consists of identifying both the information to be accumulated (memory) and a method to translate the memory into knowledge (learning). Determining the memory to be accumulated depends on the objective behind learning. Translating memory into knowledge can be done by employing various approaches. Incorporating a ML algorithm is an accepted approach due to its modularity. In this case, a supervised ML algorithm (specifically, IDT) will be used. To construct the IDT, ID3 [21] algorithm will be used.

In order to construct an IDT, a labeled training sample needs to be available to help in creating the tree. Since the problem of interest is of conceptual nature, a training sample is not available. Therefore, an initial percentage of Meta-RaPS' iterations will be used to construct the decision tree. This percentage will be known as tree% (or $t\%$). The attributes and labels that make up the instances of the training set represent the memory to be accumulated.

The proposed attributes are: the overall distance of the recently constructed solution, the number of Meta-RaPS iterations depleted so far, and the current learning policy. The overall distance is used to identify a superior solution, which is defined as the solution that exceeds the best solution found so far by $s\%$.

The proposed labels, which represent the classes used or the end nodes in a decision tree, represent different learning policies to be used in the next Meta-RaPS iteration. The proposed labels are: follow original logic, improve superior solution, or update the values of Meta-RaPS parameters. These labels allow Meta-RaPS to assess its current status and react appropriately.

4.5. Demonstration

For the initial *t%* iterations, Meta-RaPS is to execute the original construction and (possibly) improvement phases. The solutions generated make up the instances for the training sample. The training sample will be used as input for the ID3 algorithm, which will produce an IDT. Since the values of the attributes in the training sample can vary due to the randomness in Meta-RaPS, the layout of the IDT will not be known in advance. Figure 2 below represents a possible IDT. In this case, ID3 identified the number of Meta-RaPS iterations depleted so far as the attribute with the most information gain, which allowed to be the root node.

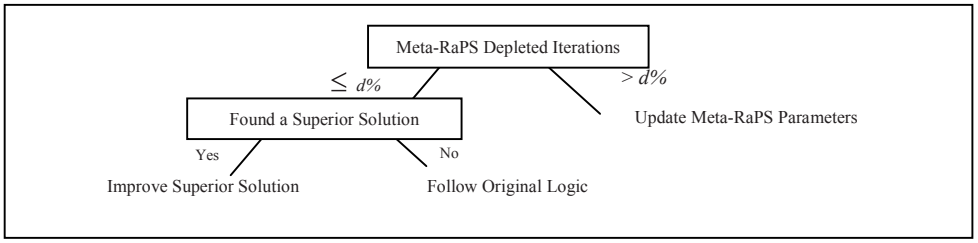

Fig. 2. Potential IDT Layout

In the iteration following the generations of the IDT, Meta-RaPS will execute the construction phase and possibly the improvement phase. At the end of every iteration, the IDT will be consulted to determine the policy to be employed next. The IDT in Figure 2 allows Meta-RaPS to follow the original logic while the iterations are depleted and a superior solution is not found. Once a superior solution is found, the policy to be employed in the next iteration will focus on perturbing the superior solution and applying local search heuristics. This follows the core idea behind the Iterative Local Search metaheuristic [26], which assumes that an algorithm used to improve a solution will continue to improve it if repeated. Thus, when a superior solution is found, the next iteration will only execute the improvement phase. This policy forgoes the multi-start nature of Meta-RaPS in an effort to focus on improving the superior solution. When iterations of Meta-RaPS are depleted, then Meta-RaPS' parameters will be updated. Updating Meta-RaPS iterations allows it to be adaptive [27]. The parameters to be updated are *r%* and *i%*. The restriction percentage is to be decreased, which allows for more arcs to be considered in the available list. The improvement percentage is to be increased, which would allow for more constructed solutions to be improved.

5. Conclusion

The main objective of this paper is to continue advancing the performance of metaheuristic methods, specifically Meta-RaPS. The concepts described represent an ongoing effort that incorporates memory and supervised machine learning into Meta-RaPS. The goal is to achieve better performance by employing an IDT to learn the status of Meta-RaPS and react by applying the appropriate policy.

Ongoing implementation will help evaluate the proposed concept. Several factors will be examined and might impact the proposed approach. One of the factors focuses on examining the effectiveness of the selected attributes and labels used by the IDT. The attributes and labels represent the scope of learning. The proposed concept focuses on learning about the overall status of Meta-RaPS performance. This is accomplished with attributes such as the overall distance of the constructed solution, the number of iterations depleted, and the current learning policy. Depending on the status of Meta-RaPS, the decision tree suggests a policy that impacts future iterations by allowing for more intensification or diversification. However, another approach could focus on selecting attributes and labels that focus on the decisions being made during constructing a solution. In the CVRP application used in this paper, the attributes may include the number of routes in a solution and the sequence of destinations in each route. These attributes can be used to assess the candidate arcs to be used in merging the routes. The labels in this case would determine if an arc is to be used or dismissed.

To further evaluate the proposed approach, numerical experiments will be performed. The input for these experiments will comprise of CVRP instances available in the literature and the results will be analyzed and compared with Meta-RaPS with different learning mechanisms and with other metaheuristics.

References

1. Taillard E. D., Gambardella L. M., Gendreau M., Potvin J. Y., "Adaptive Memory Programming, A Unified View of Metaheuristics", European Journal of Operational Research 135, pp. 1-16, 2001

2. DePuy G.W., Moraga R.J., Whitehouse G.E. "Meta-RaPS: a simple and effective approach for solving the traveling salesman problem", Transportation Research Part E, Vol. 41, p. 115–130, 2005

3. Glover F., "Future Paths for Integer Programming and Links to Artificial Intelligence". Computers and Operations Research 13 (5), pp. 533–549, 1986

4. Zhu X., Goldberg A. B. , "Introduction to Semi-Supervised Learning". Morgan and Claypool Publishers, 2009

5. Kotsiantis S.B., "Supervised Machine Learning: A Review of Classification Techniques". Informatica 31, pp. 249-268, 2007

6. Lan G., DePuy, G.W. , "On the effectiveness of incorporating randomness and memory into a multi-start metaheuristic with application to the Set Covering Problem", Computer & Industrial Engineering, Vol. 51, p. 362-374, 2006

7. Mirza A. A. B., "Improved Meta-RaPS approach with learning concepts for solving capacitated vehicle routing problem", M.S. thesis, Northern Illinois University, DeKalb, IL, 2011

8. Arin A., Rabadi G., "Memory and Learning in Metaheuristics. Artificial Intelligence", Evolutionary Computing and Metaheuristics. pp. 435-476, 2013

9. Dantzig, G.B., Ramser, J.H., "The Truck Dispatching Problem". Management Science 6 (1), pp. 80–91, 1959

10. Laporte, G., (1992), "The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms," European Journal Operationial Res. 59, 345-358 (1992).

11. Toth P., Vigo D.,"TheVehicle Routing Problem", Monographs on Discrete Mathematics and Applications, SIAM, Philadelphia, 2002

12. DePuy G.W., Whitehouse G.E., "A simple and effective heuristic for the multiple resource allocation problem, International Journal of Production Research", Vol. 32, 424-31, 2001.

13. Arcus, A. L., COMSOAL: a computer method of sequencing operations for assembly lines. Int Jr of Production Research, 4, 259–277, 1966.

14. R.J. Moraga, G.W. DePuy & G.E. Whitehouse, "Metaheuristics: A Solution Methodology for Optimization Problems. In: A.B. Badiru (ed.), Handbook of Industrial and Systems Engineering", CRC Press, FL, 2006

15. R.J. Moraga, "Meta-RaPS: An Effective Solution Approach for Combinatorial Problems", Ph.D. thesis, University of Central Florida, Orlando, FL, 2002.

16. R.J. Moraga, G.W. DePuy & G.E.Whitehouse, "Meta-RaPS approach for the 0–1 multidimensional knapsack problem, Computers & Industrial Engineering, Vol. 48, No. 2, 83-96, 2005

17. G. Rabadi, R.J. Moraga, & A. Al-Salem, "Heuristics for the unrelated parallel machine scheduling problem with setup times", Journal of Intelligent Manufacturing, Vol. 17, 85-97, 2006

18. S. Hepdogan, R.J. Moraga, G.W. DePuy & G.E. Whitehouse, "A Meta-RaPS For The Early/Tardy Single Machine Scheduling Problem", International Journal of Production Research, Vol. 47, No. 7, 1717-1732, 2009.

19. C. Garcia and G.Rabadi, "A Meta-RaPS algorithm for spatial scheduling with release times". Int. J. Planning & Scheduling, Vol. 1, Nos. 1/2, 19-31, 2011.

20. Kaplan S., Rabadi G., "A Simulated Annealing and Meta-RaPS Algorithms for the Aerial Refueling Scheduling Problem with Due Date-to-Deadline Windows and Release Time", Engineering Optimization, V. 45, No. 1, P.67-87, 2013.

21. Quinlan J. R., "Learning efficient classification procedures and their application to chess end games," in Machine Learning: An ArtiJcial Intelligence Approach, R. S . Michalski, J. G. Carbonell, and T. M. Mitchell, Eds. Palo Alto, CA: Tioga Publishing, 1983, pp. 463-482.

22. Shannon, C. E., "A mathematical theory of communication". Bell System Technical Journal, 27, 379-423, 1948

23. Utgoff, P.E.. "Incremental induction of decision trees". Machine Learning, 4, 161-186, 1989

24. Clarke, G., and Wright, J., "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," Operations Research, Vol. 12,No. 4, pp. 568-581, 1964.

25. Leopold C., "Arranging program statements for locality on the basis of neighbourhood preferences". Int J of Approximate Reasoning 19, pp. 73-90, 1998

26. Gendreau M., Potvin, J., "Handbook of Metaheuristics", Chapter 12 (p. 363-397), Springer, New York, 2010

27. Smith, J.E., "Self-Adaptation in Evolutionary Algorithms for Combinatorial Optimisation", Adaptive and Multilevel Metaheuristics, pp. 31-57. Springer, Berlin, Heidelberg, New York, 2008.