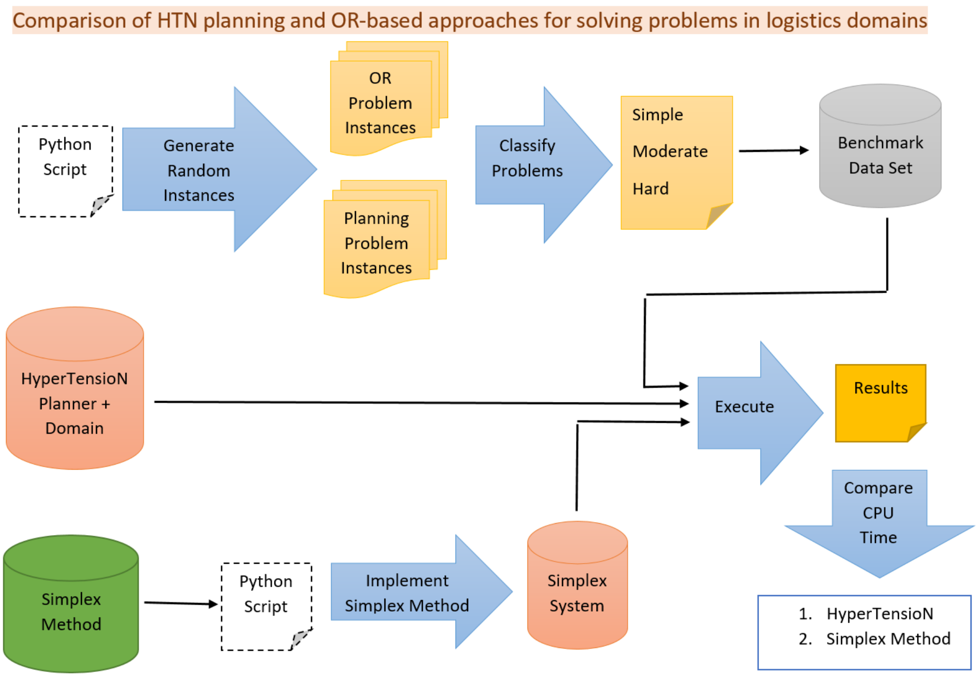
Comparison of HTN Planning and OR-based Approaches for Solving Problems in Logistics Domains

M.H.A. Sandamali and D.D.A. Gamini

Department of Computer Science, University of Sri Jayewardenepura, Nugegoda, Sri Lanka

Date Received: 04-01-2024 Date Accepted: 29-06-2024



Abstract

Logistics is a day-to-day real-life situation leading to a very large domain. Among the various kinds of logistics problems we encounter are those pertaining to transportation, product inventories and location. Traditionally, Operations Research (OR) techniques are used most often and are popular candidates for solving logistics problems. With the development of Hierarchical Task Network (HTN) planning algorithms, there has been a recent trend to use HTN planners to address planning problems in general. It is also envisaged that HTN planners could solve logistics problems; yet, no proof or evidence could be found as to whether this approach is more efficient or suitable to handle logistics problems. In this paper, a comparison is done to ascertain whether HTN planning could be more efficient than OR-based approaches in solving logistics problems. Results revealed that HTN planning outperforms the traditional OR-based approach, especially in solving moderate-sized and more difficult harder problems. Furthermore, HTN planning is capable of solving very large scale problems, whereas OR based approach was unable to handle them within the same computational resource limitations.

Keywords: *Logistics, Operations research, Hierarchical task network, Transportation simplex method, HyperTensioN*

**Correspondence: gamini@sjp.ac.lk*

1. Introduction

1.1 Operations research

Operations research (OR) was initiated during World War II in Great Britain as a method of support for the military. Operations research groups began using radar technology and its applications to solve problems in military conflicts during 1936 (Gass, 1994); they were then supported to focus on logistics, modeling, and planning at the end of World War II. Operations researchers started to solve operations problems instead of military problems, and then OR was applied to business problems as well, after which, the business saw an increase in profits. As a result, these OR techniques were accepted to solve problems in the business area (Horvath, 1955). Operations research is the application of similar ideas to more important and challenging issues involving the operations of systems, including network of machines and organizations. Mathematical strategies are needed to be utilized when making these decisions using OR (Johnes, 2015). Descriptive, case-control, and retrospective or prospective cohort analyses are the three primary types of operations research (Zachariah et al., 2009).

Standard OR techniques include linear programming, queueing theory, game theory, inventory control models, simulation, goal programming and transportation simplex method (Hash, 2011), the last of which is used in this research to solve transportation routing problems. Operations researchers have studied transportation and logistics problems for a quite long time. The first contribution (Schrijver, 2002) is a system of solutions for the transport of salt, cement, and other things from sources to destinations along the railway system. In newspaper distribution, to minimize delivery costs while limiting overall delivery delay time, it is crucial to ascertain the most effective newspaper delivery routes and schedules, as well as the distribution of newspaper agents. In order to solve this problem, the researchers looked into the regret distance method, modified saving algorithm, weighted saving algorithm, urgent route first algorithm, and modified urgent route first algorithm (Song et al., 2002). The rapid development of mathematical programming techniques together with the advances in computer hardware and software made a greater impact on the advances in operations research (Speranza, 2018).

1.2 Logistics problems

The logistics industry encompasses a broad spectrum of real-world problems and a vast array of challenges including those related to transportation problems, inventory, production and location problems. Organizing people to fight the fire, organizing people to accomplish military tasks, and organizing people to respond to natural disasters are a few examples of problems in the logistics domain, the characteristics of which are unique and generic as every problem has a unique aspect and requires a unique answer. Logistics problems are related to the pragmatic arrangement required to ensure the success of a complex plan involving several individuals and equipment. There is a long history of practical problems with logistics and transportation, under which items are produced in one or more factories and are delivered to one or more warehouses. Logistics problems are arisen in this situation when a company seeks to operate with lower expenses and higher profits (Yang et al., 2008).

1.3 Hierarchical task network

AI planning approaches are used to solve logistics problems. Dynamic Analysis and Replanning Tool (DART) is one such system deployed by U.S. forces during the Persian Gulf War in 1991 for automating logistics planning and scheduling for the transportation of military resources such as vehicles, cargo, and personnel. The system could handle a large number of vehicles and cargo up to 50,000 (Cross and Walker, 1994). In a separate study, a planning model that is to be incorporated into a logistical decision support system for natural disasters has been developed. The dynamic time-dependent transportation problem, which must be repeatedly handled at predetermined intervals, is addressed by the model. The plan for mixed pick-up and delivery times for vehicles as well as the ideal amounts and kinds of loads picked up and delivered on various routes are all provided in the plan (Ozdamar et al., 2004).

*Correspondence: gamini@sjp.ac.lk

The hierarchy is a widely utilized framework for comprehending ideas in the universe. Real-world tasks usually come with a built-in hierarchical structure; computational tasks, military tasks, and administrative tasks are just a few examples. It would be a waste of time to construct plans from individual operators, since in-built hierarchies can be used to avoid the exponential explosion of planning. The branch of Artificial Intelligence (AI) planning that represents and manages hierarchies is referred to as Hierarchical Task Network planning (Erol et al., 1996), the objective of which is to generate a sequence of actions performing a task. HTN planning includes a set of methods, each of which tells how to decompose a task into smaller tasks. Planning proceeds in this way by using methods and decomposing tasks recursively into smaller tasks until it reaches primitive tasks that could be directly performed by operators.

HTN planning uses hierarchical task network to automate the planning process to reduce the search space when finding a solution to a planning problem. An initial state is transformed into a goal state by applying ordered available actions in traditional methods. When finding a solution, there need to be lots of possible actions at every decision point, so the search space is immense (Ghallab et al., 2004). The best way to understand HTN planning is to compare it to its predecessor, STRIPS-style planning, which is quite similar to HTN planning in terms of both world representations and actions (Georgievski and Aiello, 2015). HTN planners and STRIPS-style planners differ from one another in terms of what they plan for and how they plan for it. Finding operators with the desired effects and stating their preconditions as sub-goals is how planning proceeds forward. On the other hand, one of the motivations for HTN planning was to close the gap between operations research techniques for project management and scheduling, and AI planning techniques (Tate, 1977). A recent research shows that using planning graphs approach, originally developed for action based planning, can greatly enhance HTN planning performance (Lotem et al., 1999) by combining Graphplan (Blum and First, 1995) style planning graph creation with HTN style problem reduction.

We have selected the HyperTensioN planning algorithm in this research for comparison. This approach of planning makes use of a three-stage compiler designed to support optimizations in multiple domain description languages. New domain description languages can be supported with ease due to the flexibility provided by the front and back-end modules and the middle-end pipeline makes it possible to conduct a variety of transformation and analysis tools to be executed before planning (Magnaguagno et al., 2020).

1.4 Transportation problem

Transportation science emerged in the 1960s and 1970s. Back then, traffic and public transportation were called transportation. Following the 1980s, several research studies led to the development of rail, sea, and air transportation. Thereafter, transportation was expanded to encompass passenger and freight transportation. In 1990s, logistics began to arise with operations and then later evolved to supply chain management. Between 2000 and 2010, a large number of real-life applications were covered by transportation and logistics (Speranza, 2018).

2. Methodology

2.1 Mathematical model

Transportation models have been extensively used in business problems such as, control and design of manufacturing plants, determining sales territories, and locating distribution centers and warehouses, etc. Tremendous cost savings have been achieved through the efficient transportation of goods (these will be called the packages in our research) from sources to destinations. Trucks are used to transport packages or deliver packages from sources to destinations. In general, a transportation problem is specified by the following information:

1. A set of m sources from which packages are delivered. Source i can supply at most s_i packages.

*Correspondence: gamini@sjp.ac.lk

2. A set of n destinations to which the packages are delivered. Destination j must receive at least d_j packages.
3. Each package from source i and delivered to destination j incurs a variable transportation cost of c_{ij}.

A transportation table can be constructed with the pertinent data as shown in Figure 1.

	Destination 1	Destination 2	.....	Destination n	SUPPLY
Source 1	c_{11}	c_{12}		c_{1n}	s_1
Source 2	c_{21}	c_{22}		c_{2n}	s_2
.....					
Source m	c_{m1}	c_{m2}		c_{mn}	s_m
DEMAND	d_1	d_2		d_n	

Figure 1: Prototype of transportation problem

The objective of the transportation model is to determine x_{ij}, the number of packages delivered from source i to destination j, over all routes (i, j) so as to minimize the total cost of transportation. If total supply equals total demand, then the problem is said to be a balanced transportation problem. Otherwise, it is referred to as an unbalanced transportation problem. When there is more supply than demand, we can create a dummy destination with a demand equal to the surplus supply to balance the transportation problem. Shipments to the dummy destination have a unit transportation cost of zero since they are not actual shipments. On the other hand, there is no feasible solution to the problem if the total supply falls short of the total demand. Sometimes it is preferable to give in to the prospect of unfulfilled demand in order to resolve this issue. In these circumstances, unfulfilled demand is frequently penalized, and the problem is balanced by adding a dummy source. Hence, it is sufficient to look at a way of solving a balanced transportation problem, which is formulated as below.

$$\text{Minimize} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij}$$

Subject to

$$\sum_{j=1}^{n} x_{ij} = s_i \quad ; i = 1,2,\ldots,m \qquad \text{Supply constraints}$$

$$\sum_{i=1}^{m} x_{ij} = d_j \quad ; j = 1,2,\ldots,n \qquad \text{Demand constraints}$$

$$x_{ij} \text{ (an integer)} \geq 0 \qquad ; i = 1, 2, \ldots, m; \; j = 1, 2, \ldots, n$$

2.2 Transportation simplex method

The transportation simplex method is an iterative method that requires an initial feasible solution to start with. A number of methods are available to find this initial feasible solution to a balanced transportation problem, and we have used the Northwest corner method in our research.

2.3 HyperTensioN planner

HTN planning approach is an automated planning technique that allows the links between actions to be stated as a hierarchy. This approach defines planning problems by providing a set of

*Correspondence: gamini@sjp.ac.lk

tasks, which might be primitive, compound, or goal. The basic algorithm for HTN planning is shown in Figure 2. HyperTensioN is a Ruby-written three-stage compiler (Magnaguagno et al., 2020) (Figure 3) for HTN planning, whose original idea was to convert conventional planning instances into hierarchical planning instances automatically. Before compilation into the target representation, the Hype tool allows various middle-ends to execute, even repeatedly, controlling module execution at each level. The HyperTensioN TFD (Ghallab et al., 2004) planner completes the HTN compiler output completing this pipeline with the plan output.

```
Algorithm planning(list tasks)
  return empty plan if tasks = empty
  current_task ← shift element from tasks
  if current_task is an Operator
    if applicable(current_task)
      apply(current_task)
      plan ← planning(tasks)
      return current_task.plan if plan
    end
  else if current_task is a Method
    for methods in decomposition(current_task)
      for subtasks in unification(methods)
        plan ← planning(subtasks.tasks)
        return plan if plan
      end
    end
  end
  return Failure
end
```

Figure 2: Basic algorithm for HTN planning (Magnaguagno et al., 2020)

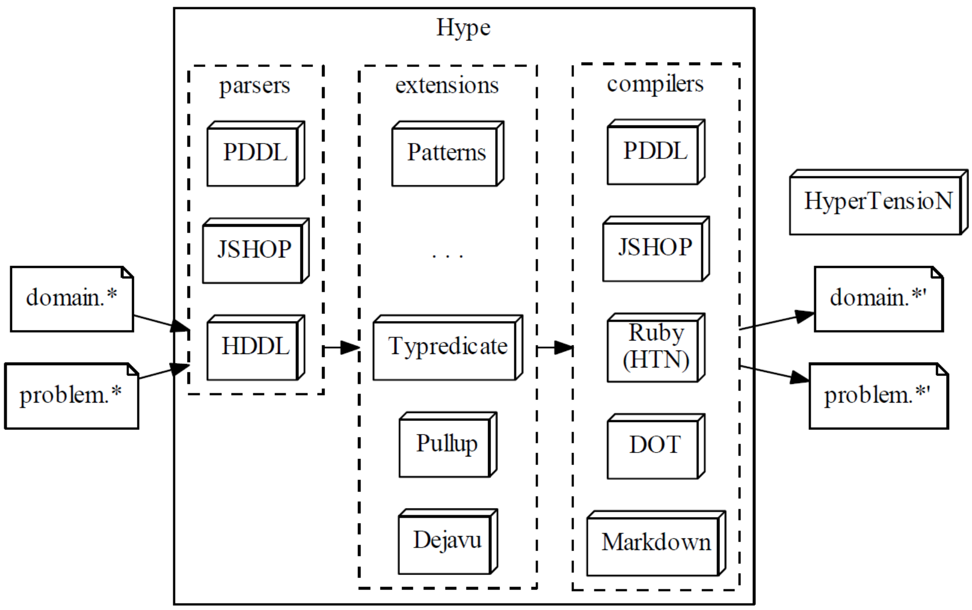


Figure 3: HyperTensioN three-stage compiler (Magnaguagno et al., 2020)

*Correspondence: gamini@sjp.ac.lk

2.4 Problem generator

A problem generator was implemented using Python to generate OR and HTN instances of varying degrees of complexity. Since unbalanced transportation problems can be handled by adding an extra source/destination, we have generated only the balanced problems. First, OR problem instances are generated, and then using the same parameters, HTN instance files are generated. These problem instances are then executed using the transportation simplex method and HyperTensioN planner, and the outputs of both methods are collected separately into individual text files.

3. Results

We have generated 5000 problem instances with 5 instances of the same problem size. We started by generating problem instances with 2 sources and 2 destinations, randomly assigning a number of packages to each source, and then randomly assigning a destination for each package at each source. We have repeated the above process by gradually increasing the number of sources, destinations, and packages ending up at 1,000 sources.

Problem instances are divided into three categories, namely simple, moderate and hard problems, based on problem size and hence ensuring the complexity of the problem. The product of the number of sources and the number of destinations, which gives the size of the cost matrix in terms of the number of elements for transportation simplex method, is considered to be the problem size. The sizes up to 900 being classified as simple problems, sizes 900 – 2,500 as moderate, and sizes 2,500 – 10,000 as hard problems.

CPU time, without including I/O overhead, is used as the performance metric for this research. CPU times have been recorded for each problem instance executed under both approaches and averaged over the five instances of the same problem size. All experiments were run on an 8th generation Intel dual core-i3 processor running at 2.20 GHz with 8 GB RAM. We have imposed a time limit of 30 minutes in our experimental set up.

3.1 Simple problem instances

Figure 4 shows the average CPU times in seconds to solve all the problem instances in the simple problems category by transportation simplex method and by HyperTensioN planner. It could be seen from this graph that CPU times are similar for around first half of the problem instances in this category but for the rest, the HyperTensioN planner gave less CPU time compared to those by transportation simplex method.

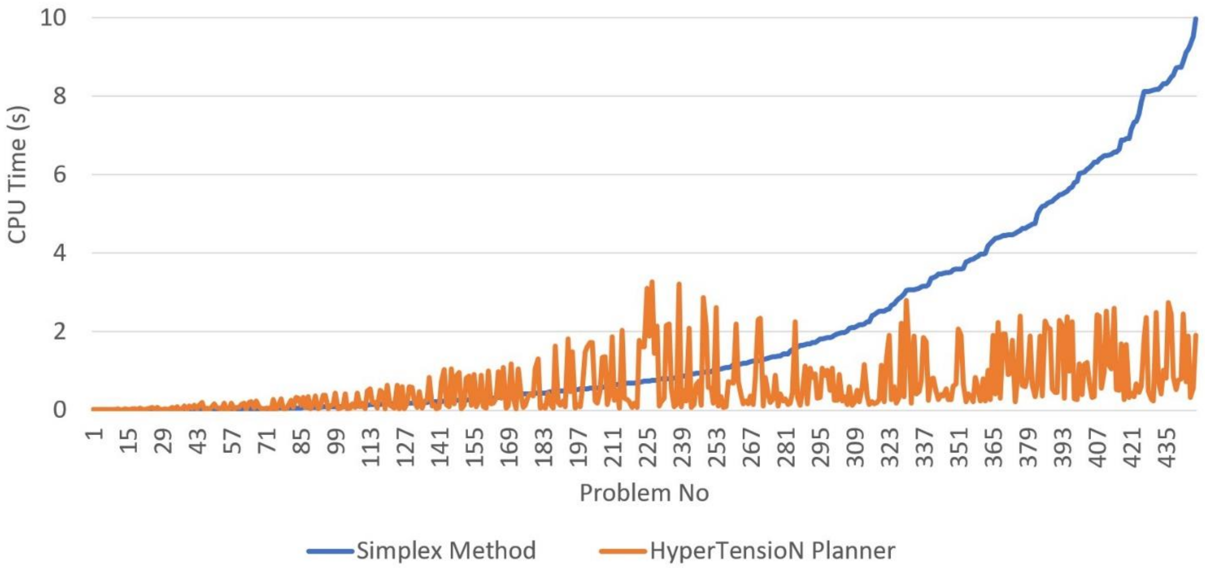


Figure 4: Average CPU times for simple problems

*Correspondence: gamini@sjp.ac.lk

3.2 Moderate problem instances

Figure 5 shows the average CPU times in seconds to solve all the problem instances in the moderate problems category by transportation simplex method and by HyperTensioN planner. It is evident from the graph in the figure that CPU times for the HyperTensioN planner are way below for those by transportation simplex method for all the problem instances in the moderate problems category, making it clear that HyperTensioN planner outperforms the transportation simplex method as the complexity of the problem increases.

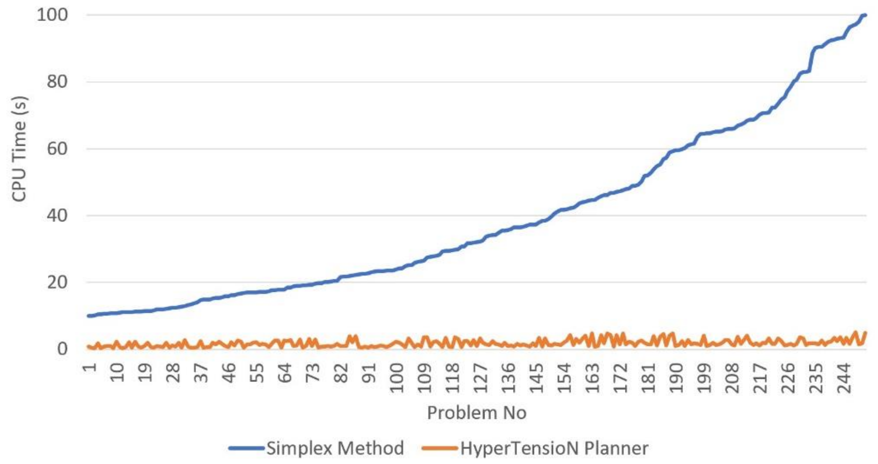


Figure 5: Average CPU times for moderate problems

3.3 Hard problem instances

The average CPU times to solve all the hard problem instances by the two methods are shown in Figure 6. Results revealed that some hard problems instances could not be solved by the transportation simplex method within our computational resource limitations. These problem instances are collected and shown in Table 1 together with CPU times by HyperTensioN planner. It could be seen from Table 1 that the HyperTensioN planner took less than 10 seconds to solve these hard problems. It is clear from the figure that CPU times for the HyperTensioN planner are way below the CPU times by transportation simplex method for all hard problem instances.

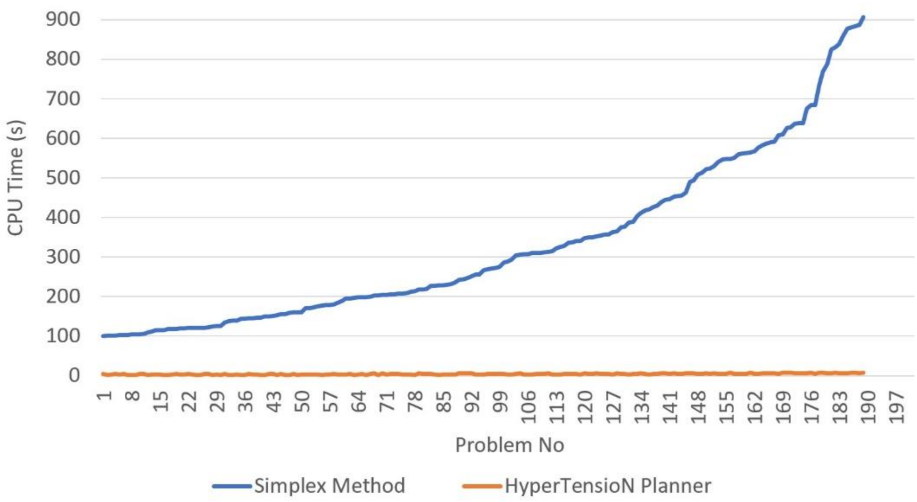


Figure 6: Average CPU times for harder problems

**Correspondence: gamini@sjp.ac.lk*

3.4 Very hard problem instances

Since some of the hard problems could not be solved by the transportation simplex method, we have additionally generated a set of 200 very hard problem instances to investigate the capability of the HyperTensioN planner for scaling up. Very hard problems generated include up to 1400 sources and thousands of packages. These hardest problems could be solved by the HyperTensioN planner in less than 20 seconds, whereas the transportation simplex method could not solve any of them within our computational resource limitations.

Table 1: Average CPU times in seconds for harder problems

Problem instance number	CPU time (s) to solve by HyperTensioN planner
987	7.69
988	7.03
989	7.72
990	7.95
991	9.51
992	8.21
993	9.12
994	8.06
995	8.11
996	8.19
997	8.31
998	8.61
999	9.05

4. Discussion

There are various approaches for solving logistics problems; the two most well-known ones are operations research and hierarchical task network planning. However, no literature could be found on the research done to compare these two approaches to determine which is better suited for solving logistics problems. Our research addressed this problem. In contrast to operations research techniques, hierarchical task network planning proves to be the most effective method for solving logistics problems, according to our investigation into this issue. Several operations research techniques are available to address logistics problems; however, in this research, we exclusively employed the transportation simplex method since it was specifically developed to solve transportation-related logistics problems, which is the type of logistics problems we examined in this paper. Further, a variety of hierarchical task network planning algorithms exist; nevertheless, the HyperTensioN planner was chosen for our research primarily due to the fact that it won the first place in the International Planning Competition in 2020.

Results presented in the previous section revealed that when the complexity of the problem increases by increasing the number of sources and destinations, the transportation simplex method fails to find the solution within the computational resource limitations. Increasing the number of sources/destinations increases the dimension of the cost matrix used by the transportation simplex method requiring the method to manipulate bigger matrices, and that might be where the computational resources are exhausted for the transportation simplex method. Further, the simplex method could not solve any of the 200 very hard problem instances we generated, whereas the HyperTensioN planner was able to solve all of them with the hardest problem solved in less than 20 seconds.

5. Conclusions

Based on our experiments we have found that the HyperTensioN planner outperformed the transportation simplex method in solving transportation problems, especially the moderate and hard

*Correspondence: gamini@sjp.ac.lk

problems. The transportation simplex method failed to find the solutions to some hard problems within our computational resource limitations. Furthermore, it could not solve any of the 200 bigger problems we have generated, which indicates that the simplex method is unable to scale up to larger problems whereas the HyperTensioN planner solved them in less time within the same computational resources. We conclude that HTN planning is more appropriate for solving larger transportation problems compared to OR-based simplex approaches.

Finding the best way to solve logistics problems is the aim of this research, so that we may be able to find the optimal solution with less cost and in less time. Moving forward with this research and extending it to include other logistics problems, we can hope to find the best method to solve overall logistics problems, since we have considered only the transportation-related logistics problems.

References

Blum, A., Furst, M., 1995. Fast planning through planning graph analysis, 14th International Joint Conference on Artificial Intelligence, pp 1636-1642.

Cross, S.E., Walker, E., 1994. Dart: applying knowledge-based planning and scheduling to crisis action planning. Intelligent Scheduling by M. Zweben and M.S. Fox. Morgan Kaufmann.

Erol, K., Hendler, J., Nau, D.S., 1996. Complexity results for HTN planning. Annals of Mathematics and Artificial Intelligence. 18(1), 69–93.

Gass, S.I., 1994. Public sector analysis and operations research/management science, Handbooks in Operations Research and Management Science, 6, 23–46.

Georgievski, I., Aiello, M., 2015. HTN planning: Overview, comparison, and beyond. Artificial Intelligence. 222, 124–156.

Ghallab, M., Nau, D., Traverso, P., 2004. Automated Planning: Theory and Practice. Elsevier, 2004.

Hash, P.M., 2011. The universal teacher, by J.E. Maddy and T.P. Giddings (1923). Journal of Research in Music Education. 58(4), 384–410.

Horvath, W.J., 1955. Some thoughts on operations research on municipal operations. Journal of the Operations Research Society of America. 3(3), 339–340.

Johnes, J., 2015. Operational research in education. European Journal of Operational Research. 243(3), 683–696.

Lotem, S., Dana, S.N., Hendler, J.A., 1999. Using planning graphs for solving HTN planning problems, 16th National Conference on Artificial Intelligence, Florida, USA, pp 534-540.

Magnaguagno, M.C., Meneguzzi, F.R., De Silva, L., 2020. Hypertension: A three-stage compiler for planning, 30th International Conference on Automated Planning and Scheduling (ICAPS), France.

Ozdamar, L., Ekinci, E., Kucukyazici, B., 2004. Emergency logistics planning in natural disasters. Annals of Operations Research, 129(1), 217– 245.

Russell, S.J., Norvig, P., 2022. Artificial Intelligence: A Modern Approach, Fourth Edition. Pearson.

Schrijver, A., 2002. On the history of the transportation and maximum flow problems. Mathematical Programming, 91, 437–445.

Song, S.H., Lee, K.S., Kim, G.S., 2002. A practical approach to solving a newspaper logistics problem using a digital map. Computers & Industrial Engineering, 43(1-2), 315–330.

Speranza, M.G., 2018. Trends in transportation and logistics, European Journal of Operational Research, 264(3), 830–836.

Tate, A., 1977. Generating project networks, 5th International Joint Conference on Artificial Intelligence Volume 2, 888–893.

Yang, Y., Jin, W., Hao, X., 2008. Car rental logistics problem: A review of literature, 2008 IEEE International Conference on Service Operations and Logistics, and Informatics, Beijing, China, pp 2815–2819.

*Correspondence: gamini@sjp.ac.lk

Zachariah, R., Harries, A.D., Ishikawa, N., Rieder, H.L., Bissell, K., Laserson, K., Massaquoi, M., Van Herp, M., Reid, T., 2009. Operational Research in Low-income Countries: What, Why, and How? The Lancet Infectious Diseases, 9(11), 711–717.

*Correspondence: gamini@sjp.ac.lk